Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An efficient multi-scale Poisson solver for the incompressible Navier–Stokes equations with immersed boundaries

Giuseppe Bonfigli [*], Patrick Jenny

Institute of Fluid Dynamics, Sonneggstrasse 3, ETH Zurich, Zurich CH-8092, Switzerland

ARTICLE INFO

Article history:
Received 1 August 2008
Received in revised form 17 February 2009
Accepted 21 March 2009
Available online 1 April 2009

Keywords:
Multi-scale
Finite-volume
Incompressible flows
Navier–Stokes
Immersed boundary
Immersed interface

ABSTRACT

The iterative-multi-scale-finite-volume (IMSFV) procedure is applied as an efficient solver for the pressure Poisson equation arising in numerical methods for the simulation of incompressible flows with the immersed-interface method (IIM). Motivated by the requirements of the specific IIM implementation, a modified version of the IMSFV algorithm is presented to allow the solution of problems, in which the varying coefficient of the elliptic equation (e.g. the permeability of the medium in the context of the simulation of flows in porous media) varies over several orders of magnitude or even becomes zero within the integration domain. Furthermore, a strategy is proposed to incorporate the iterative procedure needed by the IIM to converge out constraints at immersed boundaries into the iterative IMSFV cycle. No significant deterioration of performance of the IMSFV method is observed with respect to cases, in which no iterative improvement of the boundary conditions is considered.

1. Introduction

The IMSFV procedure by Hajibeygi et al. [1] was developed in the context of the simulation of flows in porous media for the solution of the elliptic problem

$$\frac{\partial}{\partial x_j}\left(\lambda\frac{\partial p}{\partial x_j}\right)=R, \tag{1}$$

where $p,\ \lambda$ and R are the pressure, the permeability of the medium and a distributed mass source, respectively (we use the Einstein convention for summation over repeated indices). If λ varies rapidly in space and spans a wide range of values, numerical handling of Eq. (1) becomes challenging.

The origin of the IMSFV method traces back to the work by Hou and Wu [2] in the Finite-Element context (multi-scale-finite-element, MSFE). Exploiting expected scale-separation features in the permeability field, only large scales of the solution are resolved on a correspondingly coarse grid, while fine scale features of the λ-field are accounted for in the definition of the basis functions for the finite-element approach. The latter are computed as well resolved numerical solutions of the original elliptic problem on small subdomains with properly approximated boundary conditions. Jenny et al. [3] preserved the basis-function concept but considered a finite-volume discretization (multi-scale-finite-volume, MSFV), whereby the derivation of the coarse-grid problem can still be interpreted as a weak non-Galerkin formulation of the governing equation. Continuing this work Lunati and Jenny [4] reconstructed the fine-grid solution superposing not only basis functions but also a second set of locally defined functions (correction functions), which account for the fine-scale features of non-homogeneities (i.e.

[*] Corresponding author. Tel.: +41 44 632 8682; fax: +41 44 632 1147.
E-mail address: bonfigli@ifd.mavt.ethz.ch (G. Bonfigli).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.032

source terms and non-homogeneous boundary conditions). In a further step, a divergence-free vector field is computed to approximate the Darcy velocity $u_j = -\lambda \partial p / \partial x_j$ (conservative flow field).

The idea of improving the solution iteratively was introduced by Durlofwsky et al. [5] in a formulation similar to that by Jenny et al. [3]. Before starting a new iteration, a fine-scale solution is reconstructed and used to define improved boundary conditions for a new set of basis functions. This leads to an improvement in the results, but convergence at the fine-grid level is not achieved. Following an alternative approach, Hajibeygi et al. [1] proposed the iterative-multi-scale-finite-volume (IMSFV) procedure as a generalization of the MSFV method. The fine scale solution is reconstructed by superimposing both basis and correction functions and used to iteratively improve the approximate boundary conditions for the latter. In addition line relaxation is applied to smooth the fine-grid solution, which is determinant to ensure stability and enhance convergence. Fast convergence to the exact fine-grid solution was achieved for a variety of problems, but performance decay was observed for cases in which the permeability varies over several orders of magnitude (e.g. the SPE10 test case in [1]).

The IIM approach by LeVeque and Li [6] can be interpreted as a refinement of the immersed-boundary method (IBM) devised by Peskin [7] for the simulation of incompressible flows around complex geometries using Cartesian grids. In IBM the influence of the immersed body is modeled by means of concentrated volume forces enforcing boundary conditions at the body surface. Successive improvements in the definition of the volume force were introduced by Goldstein et al. [8] and Fadlun et al. [9], whereby the latter are able to exactly enforce no-slip conditions for the velocity at solid walls. LeVeque and Li [6] observed that discontinuities in the flow quantities or in their derivatives necessarily occur across immersed boundaries and should be accounted for when defining discretization stencils crossing them. This led to the development of the IIM, for which one possible implementation in the Navier–Stokes context was provided by Linnick and Fasel [10] using finite differences for spatial discretization. Jumps across immersed boundaries are computed by means of one-sided differences and fourth-order accuracy is achieved for a stream-function-vorticity formulation of the Navier–Stokes equations. Also consistent with the requirements inspiring IIM is the primitive-variable approach by Peller et al. [11], where values for the velocity vector are extrapolated from nodes lying inside the flow field to nodes within the solid body, which are involved in stencils crossing the boundary (ghost nodes). In this way, the solution is smoothly prolonged into the body and no additional correction is needed for the considered second-order finite-volume discretization.

All IIM approaches, and also the procedure by Fadlun et al. [9], disrupt the regular structure of the Poisson problem to be solved either for the stream-function in [10] or for the pressure in all remaining procedures. This is a relevant issue, since the solution of the Poisson equation is by far the most costly step in all numerical procedures for incompressible flows, and the possibility of using highly efficient solvers on simple Cartesian grids might be one of the main advantages of IBM and IIM with respect to other approaches. Nevertheless little is said on this topic in the available literature. Fadlun et al. [9] suppress anomalies in the Poisson equation by tolerating a small additional error within the fractional-step method, which they use for time integration. Peller et al. [11] and Linnick and Fasel [10] use a single-grid SIP-method and an ILLU-based multi-grid, respectively. In both cases the preconditioner used to compute the pressure correction (or the stream-function correction) only considers a standard form of the Poisson equation, ignoring the presence of immersed bodies. Convergence to the solution fulfilling the boundary conditions at immersed boundaries is achieved over an iteration loop with alternating computation of pressure and velocity (or vorticity and stream function).

The main concern of this paper is to show that the IMSFV procedure can be adapted to efficiently solve the pressure Poisson equation as resulting from the IIM formulation by Peller et al. [11], which we prefer since it is not limited to the 2-d case like that by Linnick and Fasel [10]. Differently from previously used Poisson solvers, IMSFV provides a preconditioner, which approximatively accounts for boundary conditions at immersed boundaries. An iterative update of the velocity in the neighbourhood of the immersed boundary is still necessary, but the overall performance of procedure is significantly improved. Indeed, the importance of imposing accurate boundary conditions when computing the pressure correction within iterative procedures has been pointed out also by Mark and van Wachem [12].

The paper is structured as follows. The finite-volume discretization of the Navier–Stokes equations is presented in Section 2, where also the IIM implementation of immersed boundaries is described. We start Section 3 with showing that, if an iterative loop is introduced to evaluate velocities at ghost nodes, the Poisson equation for the pressure may be reduced to an elliptic problem of the kind (1), where immersed bodies appear as impermeable regions. The original IMSFV procedure and the modifications needed to handle problems involving impermeable regions are presented in Sections 3.1 and 3.2, respectively. Section 3.3 describes how the IMSFV method may be combined with the iterative handling at ghost nodes. Results from numerical tests are presented in Section 4 and concluding remarks are given in Section 5. Since the considered Navier–Stokes application requires high accuracy in the solution of the pressure equation, only the linear-solver capabilities of the IMSFV procedure is considered in this paper. Its capability of providing good approximations of the solution after one single iteration is not discussed.

2. Discretization of the Navier–Stokes equations

The numerical procedure is based on a primitive-variable formulation of the incompressible Navier–Stokes equations with finite-volume discretization on staggered grids [13]. The standard fourth-order Runge–Kutta scheme is used for time-integration and the immersed-boundary implementation reproduces that by Peller et al. [11]. We first present the discretized problem ignoring immersed boundaries, and then detail the implementation of the latter.

2.1. Governing equations and grid definition

Let $\underline{x} = (x_1\ x_2\ z_3)^T \in \mathbb{R}^3$ be the coordinate vector in a 3-d Cartesian coordinate system and $\underline{u} = (u_1\ u_2\ u_3)^T \in \mathbb{R}^3$ the velocity in the same reference system.We consider the rectangular integration domain

$$\Omega = \{\underline{x} :\ a_i < x_i < b_i\}, \quad i = 1, 2, 3;\ a_i, b_i \in \mathbb{R} \tag{2}$$

and allow an arbitrarily shaped open portion $\Omega_b \subset \Omega$ to be occupied by solid immersed bodies. The flow field has to be computed only within the region $\Omega_f = \Omega \setminus (\Omega_b \cup \partial\Omega_b)$, which is assumed to be simply connected. The notation $\partial\Omega$ is used here and in the following to identify the boundary of a set Ω.

The discrete problem is derived from the momentum equation in conservative form

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \quad \forall t,\ \forall \underline{x} \in \Omega_f,\ i = 1, 2, 3 \tag{3a}$$

and the continuity equation

$$\frac{\partial u_i}{\partial x_i} = 0, \quad \forall t,\ \forall \underline{x} \in \Omega_f. \tag{3b}$$

Dirichlet boundary conditions

$$u_i = \bar{u}_i, \quad \forall t,\ \forall \underline{x} \in \partial\Omega_f = \partial\Omega \cup \partial\Omega_b,\ i = 1, 2, 3 \tag{4}$$

are imposed at outer and inner boundaries, $\bar{u}_i$ being prescribed functions, which fulfill the compatibility condition

$$\int_{\partial\Omega \cup \partial\Omega_b} \bar{u}_i n_i dS = 0, \quad \forall t, \tag{5}$$

where $\underline{n}$ is the unit outwards-pointing normal on $\partial\Omega \cup \partial\Omega_b$. An extrapolation procedure may also be applied at outflow boundaries to evaluate the value of $\bar{u}_i$ on the basis of the solution at the previous time step.

Given the grid spacing $h_i = (b_i - a_i)/N_i,\ N_i \in \mathbb{N}$, the index triplet $\underline{\alpha} = (\alpha_1, \alpha_2, \alpha_3), \alpha_j = n_j/2$ for some $n_j \in \mathbb{Z}$, identifies the grid node located at

$$\underline{x} = \underline{x}_{\underline{\alpha}} = (\alpha_1 h_1 \quad \alpha_2 h_2 \quad \alpha_3 h_3)^T \tag{6}$$

and all quantities evaluated at that node (e.g. $p_{\underline{\alpha}}$ stands for the pressure at $\underline{x}_{\underline{\alpha}}$ while $u_{i,\underline{\alpha}}$ is the ith velocity component at the same position). Sum and multiplication by a scalar are defined for index triplets analogously to the vector-space operations in $\mathbb{R}^3$ and a partial order is given by setting $\underline{\alpha} \leqslant \underline{\beta}$, if $\alpha_i \leqslant \beta_i\ \forall i$. We also define the triplets

$$\underline{\delta}_1 = \left(\frac{1}{2},\ 0,\ 0\right), \quad \underline{\delta}_2 = \left(0,\ \frac{1}{2},\ 0\right), \quad \underline{\delta}_3 = \left(0,\ 0,\ \frac{1}{2}\right).$$

A set $\mathcal{G}$ of triplets is said to be a grid, if the following properties are fulfilled: (i) $\underline{x}_{\underline{\alpha}} \in (\Omega \cup \partial\Omega),\ \forall \underline{\alpha} \in \mathcal{G}$; (ii) if $\underline{\alpha}, \underline{\beta} \in \mathcal{G}$ and $(\gamma_j = \alpha_j \vee \gamma_j = \beta_j),\ \forall j$, then $\underline{\gamma} \in \mathcal{G}$. The boundary set $\partial\mathcal{G}$ is the set of triplets identifying the projections of nodes of $\mathcal{G}$ onto the boundary $\partial\Omega$.

The pressure p and the velocity components u_i are defined on the staggered grids $\mathcal{P}$ and $\mathcal{U}_i \cup \partial\mathcal{U}_i$, respectively, with

$$\mathcal{P} = \left\{\underline{\alpha} :\ \alpha_j \in \mathbb{Z} + \frac{1}{2},\ \forall j,\ \underline{x}_{\underline{\alpha}} \in \Omega\right\}, \tag{7a}$$

$$\mathcal{U}_i = \left\{\underline{\alpha} :\ \alpha_i \in \mathbb{Z},\ \alpha_j \in \mathbb{Z} + \frac{1}{2},\ \forall j \neq i,\ \underline{x}_{\underline{\alpha}} \in \Omega\right\}. \tag{7b}$$

The relative position of grid nodes for p and u_i in the 2-d case is sketched in Fig. 1. The control volume or cell $\Omega_{\underline{\alpha}}$ associated to the node $\underline{\alpha}$ is defined as the rectangular domain

$$\Omega_{\underline{\alpha}} = \left\{\underline{x} :\ \left(\alpha_j - \frac{1}{2}\right)h_j \leqslant x_j \leqslant \left(\alpha_j + \frac{1}{2}\right)h_j,\ \forall j\right\} \cap \Omega. \tag{8}$$

The symbols $V(\Omega_{\underline{\alpha}})$ and $A_{\underline{\alpha},i}$ denote the volume of $\Omega_{\underline{\alpha}}$ and the area of its face normal to the x_i-axis, respectively.

2.2. Finite-volume discretization

Partial derivatives at nodes lying in the inner of the integration domain are evaluated by means of the second-order central finite-difference scheme

$$\left(\frac{\partial f}{\partial x_j}\right)_{\underline{\alpha}} = \frac{f_{\underline{\alpha}+\underline{\delta}_j} - f_{\underline{\alpha}-\underline{\delta}_j}}{h_j}. \tag{9}$$

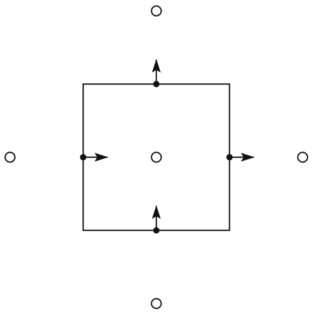


Fig. 1. Staggered grids for p (circle) u_1 (horizontal arrows) and u_2 (vertical arrows).

Volume integrals over cells $\Omega_{\underline{\alpha}}$ and surface integrals over cell faces S are defined as

$$\int_{\Omega_{\underline{\alpha}}} f dV = f_{\underline{\alpha}} V(\Omega_{\underline{\alpha}}), \quad \int_S f dS = f_{\underline{\beta}} A_{k,\underline{\beta}}, \tag{10}$$

where k is the direction normal to S and $\underline{\beta}$ is the centroid of S. Due to the additivity of the integral this defines integral over arbitrary unions of cells and faces. We also point out that, for the given definitions of discrete derivation and integration, the Gauss theorem continues to be valid also in the discrete case. Consequently the solvability condition (5) preserves its meaning also for the discretized problem, if the integral on the left-hand side is interpreted according to Eq. (10) and $\partial\Omega \cup \partial\Omega_b$ is assumed to be equal to the boundary of $\cup_{\underline{\alpha}\in\mathcal{P}_f}(\Omega_{\underline{\alpha}} \cup \partial\Omega_{\underline{\alpha}})$.

The discretized form of the Navier–Stokes equations is presented considering the explicit Euler scheme for time integration. Generalization to the Runge–Kutta scheme is straightforward. Let $\Delta t > 0$ be the time step and let the superscript (l) identify quantities evaluated at the time $t = l \cdot \Delta t$. Eq. (3a) can be written in discretized form as

$$\frac{u_{i,\underline{\alpha}}^{(l+1)} - u_{i,\underline{\alpha}}^{(l)}}{\Delta t} V(\Omega_{\underline{\alpha}}) = -P_{i,\underline{\alpha}}^{(l)} + X_{i,\underline{\alpha}}^{(l)} + Y_{i,\underline{\alpha}}^{(l)}, \quad \forall \underline{\alpha} \in \mathcal{U}_i, \tag{11}$$

where $P_{i,\underline{\alpha}}$, $X_{i,\underline{\alpha}}$ and $Y_{i,\underline{\alpha}}$ represent the contributions of the pressure gradient, of the convective term and of the viscous term, respectively, to the momentum balance in the cell $\Omega_{\underline{\alpha}}$. The Dirichlet condition (4) assumes the form

$$u_{i,\underline{\alpha}}^{(l+1)} = \bar{u}_{i,\underline{\alpha}}^{(l+1)}, \quad \forall \underline{\alpha} \in \partial\mathcal{U}_i, \tag{12}$$

whereby the values $\bar{u}_{i,\underline{\alpha}}^{(l+1)}$ may be slightly modified on some portion of $\partial\Omega$ (typically at the outflow) in order to fulfill the discrete counterpart of the compatibility condition (5). We omit the time-step superscript, indicate with $\underline{n}$ the outwards-pointing normal at the cell boundaries and set

$$P_{i,\underline{\alpha}} = \int_{\partial\Omega_{\underline{\alpha}}} p n_i dS = (p_{\underline{\alpha}+\underline{\delta}_i} - p_{\underline{\alpha}-\underline{\delta}_i}) A_{i,\underline{\alpha}}, \tag{13a}$$

$$X_{i,\underline{\alpha}} = \int_{\partial\Omega_{\underline{\alpha}}} u_i u_j n_j dS = (u_{i,\underline{\alpha}+\underline{\delta}_j} u_{j,\underline{\alpha}+\underline{\delta}_j} - u_{i,\underline{\alpha}-\underline{\delta}_j} u_{j,\underline{\alpha}-\underline{\delta}_j}) A_{j,\underline{\alpha}}, \tag{13b}$$

$$Y_{i,\underline{\alpha}} = \int_{\partial\Omega_{\underline{\alpha}}} \frac{1}{Re} \frac{\partial u_i}{\partial x_j} n_j dS = \frac{1}{Re} \left[\left(\frac{\partial u_i}{\partial x_j} \right)_{\underline{\alpha}+\underline{\delta}_j} - \left(\frac{\partial u_i}{\partial x_j} \right)_{\underline{\alpha}-\underline{\delta}_j} \right] A_{j,\underline{\alpha}}. \tag{13c}$$

The evaluation of $P_{i,\underline{\alpha}}$ is straightforward, since $(\underline{\alpha} \pm \underline{\delta}_i) \in \mathcal{P}$ whenever $\underline{\alpha} \in \mathcal{U}_i$. Second-order central interpolation is used to evaluate both the convecting velocity u_j and the convected component u_i at $x_{\underline{\alpha}\pm\underline{\delta}_j}$, as appearing in $X_{i,\underline{\alpha}}$:

$$u_{j,\underline{\alpha}\pm\underline{\delta}_j} = \frac{u_{j,\underline{\alpha}\pm\underline{\delta}_j+\underline{\delta}_i} + u_{j,\underline{\alpha}\pm\underline{\delta}_j-\underline{\delta}_i}}{2}, \quad u_{i,\underline{\alpha}\pm\underline{\delta}_j} = \frac{u_{i,\underline{\alpha}\pm\underline{\delta}_j+\underline{\delta}_j} + u_{i,\underline{\alpha}\pm\underline{\delta}_j-\underline{\delta}_j}}{2}. \tag{14}$$

Finally, partial derivatives in $Y_{i,\underline{\alpha}}$ are computed according to Eq. (9), except at nodes on $\partial\Omega$, where second-order skew differences are used. Second order accuracy is thus granted by the discretization of all terms. If needed, e.g. for higher Reynolds numbers, upwinding [14] might be introduced in the discretization of the convective term $X_{i,\underline{\alpha}}$.

A system of equations for $p_{\underline{\alpha}}^{(l)}$ is derived imposing the continuity Eq. (3b) in discrete form at all nodes $\underline{\alpha} \in \mathcal{P}$ for the time step $(l+1)$. In this sense $p_{\underline{\alpha}}^{(l)}$ is rather a Lagrangian multiplier for the problem (3a) constrained by (3b) than an estimation of the actual physical pressure, from which it differs by terms of first order in Δt [15,16]. The finite-volume formulation of Eq. (3b) provides

$$\int_{\partial\Omega_{\underline{\alpha}}} u_j^{(l+1)} n_j dS = \left(u_{j,\underline{\alpha}+\underline{\delta}_j}^{(l+1)} - u_{j,\underline{\alpha}-\underline{\delta}_j}^{(l+1)} \right) A_{j,\underline{\alpha}} = 0, \quad \forall \underline{\alpha} \in \mathcal{P}. \tag{15}$$

Since $\underline{\alpha} \pm \underline{\delta}_j \in \mathcal{U}_j \cup \partial\mathcal{U}_j$ whenever $\underline{\alpha} \in \mathcal{P}$, we may eliminate $u^{(l+1)}_{j,\underline{\alpha}\pm\underline{\delta}_j}$ in (15) by means of Eqs. (11) and (12), obtaining the desired set of equations:

$$\left(\frac{\partial^2 p}{\partial x_j \partial x_j}\right)^{(l)}_{\underline{\alpha}} = \frac{1}{V(\Omega_\alpha)}\left[\left(\frac{\partial X_j}{\partial x_j}\right)^{(l)}_{\underline{\alpha}} + \left(\frac{\partial Y_j}{\partial x_j}\right)^{(l)}_{\underline{\alpha}}\right] + \frac{1}{\Delta t}\left(\frac{\partial u_j}{\partial x_j}\right)^{(l)}_{\underline{\alpha}}, \quad \forall \underline{\alpha} \in \mathcal{P}, \tag{16a}$$

$$\left(\frac{\partial p}{\partial x_k}\right)^{(l)}_{\underline{\alpha}} = \frac{1}{V(\Omega_\alpha)}\left[X^{(l)}_{k,\underline{\alpha}} + Y^{(l)}_{k,\underline{\alpha}}\right] - \frac{\bar{u}^{(l+1)}_{k,\underline{\alpha}} - \bar{u}^{(l)}_{k,\underline{\alpha}}}{\Delta t}, \quad \forall \underline{\alpha} \in \partial\mathcal{P}, \tag{16b}$$

where x_k is the direction normal to the boundary at each point $\underline{\alpha} \in \partial\mathcal{P}$ and the values $\bar{u}^{(l)}_{k,\underline{\alpha}}$ and $\bar{u}^{(l+1)}_{k,\underline{\alpha}}$ in (16b) are the assigned right-hand side of Eq. (12). The derivatives $(\partial p/\partial x_j)_{\underline{\alpha}}$ on the left-hand side of (16a) can be evaluated according to Eq. (9) for all $\underline{\alpha} \in (\mathcal{P} \pm \delta_j) \setminus \partial\mathcal{P}$. Since furthermore their values at nodes $\underline{\alpha} \in \partial\mathcal{P}$ in the boundary-normal direction x_k are assigned by Eq. (16b), the second derivatives $(\partial^2 p/\partial x_j \partial x_j)_{\underline{\alpha}}$ can be computed for $\underline{\alpha} \in \mathcal{P}$, by applying the discrete derivative operator a second time.

The terms $X^{(l)}_{k,\underline{\alpha}}$ and $Y^{(l)}_{k,\underline{\alpha}}$, $\underline{\alpha} \in \partial\mathcal{P}$, in Eq. (16) are to some extent fictive. They do not appear in Eqs. (11) or (12) and cancel out with the corresponding terms in the derivatives of X_j and Y_j on the right-hand side of (16a) as soon as Eq. (16b) is used to eliminate $\partial p/\partial x_j$ at boundary points $\underline{\alpha} \in \partial\mathcal{P}$ from the Laplacian on the left-hand side of (16a). They have been introduced only to preserve the standard form of Eq. (16a) also at nodes close to the boundary, and may as well be assumed to be zero. That being said, we notice that the system (16) is equivalent to a discretized Poisson problem with Neumann boundary conditions. It is singular and admits solutions, which are then determined up to an additive constant, if and only if the discrete counterpart of Eq. (5) is fulfilled. Second order accuracy with respect to the spatial discretization is achieved for the momentum equation.

2.3. Immersed-boundary implementation

Immersed boundaries are implemented as in the IIM approach by Peller et al. [11] by extrapolating the velocity to ghost nodes inside the body. This is done using fourth-order extrapolation stencils, which take into account the Dirichlet condition (4) on $\partial\Omega_b$ and values of u_i at three further grid nodes in the flow field. For stability reasons the point next to the boundary may be skipped, if it is too close to the boundary.

We define subsets of the grids $\mathcal{P}$ and $\mathcal{U}_i$, which are needed to detail the different treatment of nodes inside and outside of the solid bodies:

$$\mathcal{P}_f = \{\underline{\alpha} \in \mathcal{P} : \underline{x}_{\underline{\alpha}} \in (\Omega_f \cup \partial\Omega_f)\}, \tag{17a}$$

$$\mathcal{P}_b = \mathcal{P} \setminus \mathcal{P}_f, \tag{17b}$$

$$\mathcal{U}_{i,f} = \{\underline{\alpha} \in \mathcal{U}_i : \underline{x}_{\underline{\alpha}} \in \Omega_f\}, \tag{17c}$$

$$\mathcal{U}_{i,b} = \mathcal{U}_i \setminus \mathcal{U}_{i,f}, \tag{17d}$$

$$\mathcal{I}_{i,P} = \{\underline{\alpha} \in \mathcal{U}_i : \exists \nu \in \{\pm 1\} \text{ with } (\underline{\alpha} + \nu\underline{\delta}_i) \in \mathcal{P}_f \wedge (\underline{\alpha} - \nu\underline{\delta}_i) \in \mathcal{P}_b\}, \tag{17e}$$

$$\mathcal{I}_{i,U} = \{\underline{\alpha} \in \mathcal{U}_{i,b} : \exists j, \exists \nu \in \{\pm 1\} \text{ with } (\underline{\alpha} + 2\nu\underline{\delta}_j) \in \mathcal{U}_{i,f}\}. \tag{17f}$$

The momentum Eq. (11) is evaluated only at nodes $\alpha \in \mathcal{U}_{i,f} \setminus \mathcal{I}_{i,P}$ within the flow field. Nodes $\underline{\alpha} \in \mathcal{I}_{i,U}$ lie within the solid body but are needed to evaluate the right-hand side of the momentum Eq. (11) (hence the notation $\mathcal{I}_{i,U}$) at some node in $\mathcal{U}_{i,f} \setminus \mathcal{I}_{i,P}$ (cf. Fig. 2). On the other hand, nodes $\underline{\alpha} \in \mathcal{I}_{i,P}$ are involved in the evaluation of the continuity equation, i.e. of the pressure Poisson Eq. (16b) (hence the notation $\mathcal{I}_{i,P}$), which they enter through the boundary condition at immersed boundaries (see Eq. (18) below). Nodes $\underline{\alpha} \in \mathcal{I}_{i,P}$ may lie either within the body or within the flow region (cf. Fig. 3). However, also in the latter case, interpolation is used to evaluate $u_{i,\underline{\alpha}}$ in place of imposing (11), since no pressure node is available on the side of $\underline{\alpha}$ occupied by the body and the computation of the pressure gradient would be numerically critical. The sets $\mathcal{I}_{i,P}$ and $\mathcal{I}_{i,U}$ are in general not disjoint.

As a general rule, u_i is extrapolated along the grid line on which the stencil requiring it operates, e.g. in direction x_j for the finite-difference stencil in (9) or in direction x_i and x_j for the first and second interpolation stencils in (14), respectively (see Figs. 2(a)–(c) and 3). If this is not possible, extrapolation in direction x_i, parallel to the extrapolated velocity component u_i, is used (see Fig. 2(d)). Since nodes in $\underline{\alpha} \in \mathcal{I}_{i,P}$ are needed in connection with the computation of the derivative $\partial u_i/\partial x_i$ from the continuity equation, the extrapolation direction at such nodes is always x_i, without exceptions.

The discrete Poisson Eq. (16a) for the pressure is imposed only at nodes $\alpha \in \mathcal{P}_f$, while the boundary conditions

$$\left(\frac{\partial p}{\partial x_i}\right)^{(l)}_{\underline{\alpha}} = \frac{1}{V(\Omega_\alpha)}\left[X^{(l)}_{i,\underline{\alpha}} + Y^{(l)}_{i,\underline{\alpha}}\right] - \frac{u^{(l+1)}_{i,\underline{\alpha}} - u^{(l)}_{i,\underline{\alpha}}}{\Delta t}, \quad \forall \underline{\alpha} \in \mathcal{I}_{i,P}, \tag{18}$$

are set to take account of immersed boundaries. Also here values $X^{(l)}_{i,\underline{\alpha}}$ and $Y^{(l)}_{i,\underline{\alpha}}$ are fictive and cancel out as soon as Eq. (18) is used to eliminate the pressure gradient in $\underline{\alpha} \in \mathcal{I}_{i,P}$ from the Laplacian in Eq. (16a). The term $u^{(l+1)}_{i,\underline{\alpha}}$ on the right-hand side of Eq. (18) has to be evaluated by extrapolation on the basis of some $u^{(l+1)}_{i,\underline{\beta}}$ at neighbouring nodes $\underline{\beta} \in \mathcal{U}_{i,f}$, which in turn depend on the solution for $p^{(l)}$ through the momentum Eq. (11). If directly enforced, Eq. (18) would massively modify the structure of

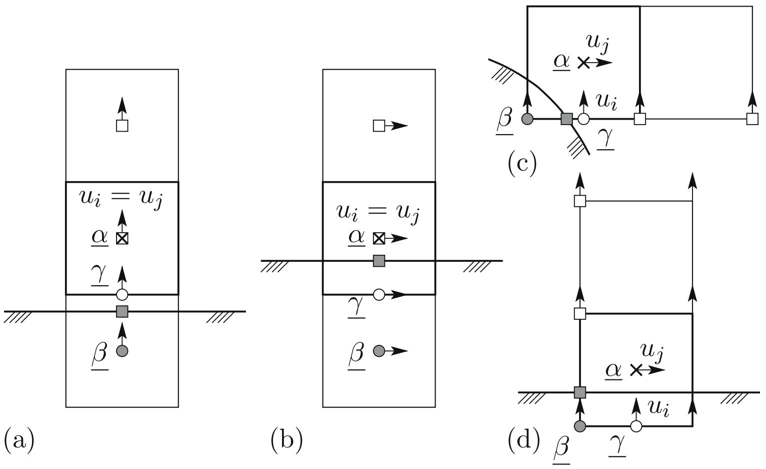


Fig. 2. Relative positions of immersed boundaries and ghost nodes $\underline{\beta} \in \mathcal{I}_{i,U}$ (filled circle). The evaluation of the momentum equation for u_j in $\underline{\alpha}$ (cross) requires the value of u_i at $\underline{\beta}$ to evaluate the same quantity (or its derivative) in $\underline{\gamma}$ (empty circle). Squares correspond to nodes included in the extrapolation stencil. Empty squares are nodes in $\mathcal{U}_{i,f}$, filled squares are boundary values assigned by (4).

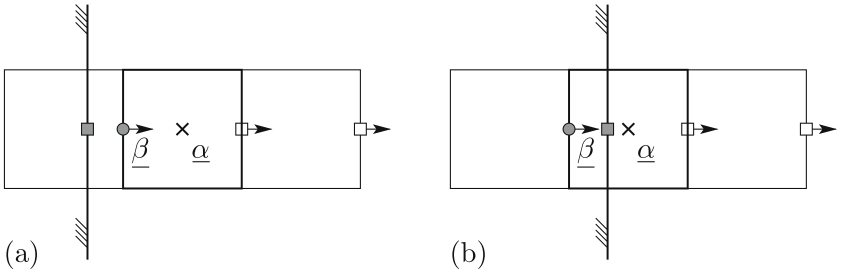


Fig. 3. Relative positions of immersed boundaries and ghost nodes $\underline{\beta} \in \mathcal{I}_{i,P}$ (filled circle). The evaluation of the pressure Poisson equation in $\underline{\alpha}$ (cross) requires the value of u_i at $\underline{\beta}$. Squares correspond to nodes included in the extrapolation stencil. Empty squares are nodes in $\mathcal{U}_{i,f}$, filled squares are boundary values assigned by (4).

the linear system (16), making impossible the application of the IMSFV procedure, or of any other standard solver. Therefore, $u_{i,\underline{\alpha}}^{(l+1)}$ is rather shifted to the right-hand side of the system, i.e. it is first evaluated approximatively and then improved iteratively (Section 3.3).

3. Computation of the pressure field with the IMSFV procedure

The linear system (16) for the discrete pressure-Poisson equation with the additional constraint (18) at immersed boundaries can be restated in the form

$$\left[\frac{\partial}{\partial x_j}\left(\lambda\frac{\partial p}{\partial x_j}\right)\right]_{\underline{\alpha}}^{(l)} = R_{\underline{\alpha}}, \quad \forall \underline{\alpha} \in \mathcal{P}, \tag{19a}$$

$$\left(\lambda\frac{\partial p}{\partial x_k}\right)_{\underline{\alpha}}^{(l)} = q_{\underline{\alpha}} \quad \forall \underline{\alpha} \in \partial\mathcal{P}, \tag{19b}$$

with

$$\lambda_{\underline{\beta}} = \begin{cases} 1, & \text{for } \underline{\beta} \in \mathcal{U}_{i,f} \setminus \mathcal{I}_{i,P}, \\ 0, & \text{for } \underline{\beta} \in \mathcal{U}_{i,b} \cup \mathcal{I}_{i,P}. \end{cases} \tag{20}$$

Thereby, the term $R_{\underline{\alpha}}$ is zero for $\underline{\alpha} \in \mathcal{P}_b$ and equal to the right-hand side of Eq. (16) for the nodes $\underline{\alpha} \in \mathcal{P}_f$ which are not neighbours of nodes in $\mathcal{I}_{i,P}$. If on the contrary $\underline{\alpha} \in \mathcal{P}_f$ and $\underline{\beta} = (\underline{\alpha} + \delta_j) \in \mathcal{I}_{i,P}$ or $\underline{\beta} = (\underline{\alpha} - \delta_j) \in \mathcal{I}_{i,P}$, then $R_{\underline{\alpha}}$ also contains the contribution of the gradient $(\partial p/\partial x_j)_{\underline{\beta}}$, assigned by Eq. (18), to the Laplacian of p on the left-hand side of Eq. (16) (remember that $\underline{u}_{\underline{\beta}}^{(l+1)}$ in Eq. (18) is treated iteratively and notice that the mentioned contribution does not appear on the left-hand side of Eq. (19a), since $\lambda_{\underline{\beta}} = 0$ for $\underline{\beta} \in \mathcal{I}_{i,P}$). Eq. (19b) is simply a restatement of the Neumann condition (16b) at the outer boundaries of the integration domain and the solvability condition for the system (19) is

$$\int_{\Omega} R dV = \int_{\partial\Omega} q dS, \tag{21}$$

which is true up to discretization errors if Eq. (5) is verified. We point out that boundary conditions at immersed boundaries are correctly accounted for in (19) but enter the system only through the coefficients λ and the right-hand side R.

The reformulation (19) of the discrete pressure Poisson equation highlights its analogy to the Eq. (1) governing flows in porous media, for the solution of which the IMSFV procedure [1] has been developed. In the rest of this section, we first present the original IMSFV procedure, then introduce the modifications needed to handle problems in which the permeability becomes zero as in (19) and (20) (impermeable regions) and finally discuss how the iterative enforcement of the boundary condition (18) can be efficiently integrated into the iterative loop of the IMSFV procedure. We skip the so called conservative reconstruction from the original IMSFV implementation [3,4,1], since it looses relevance if the iterative procedure is well converged. For technical reasons, we also slightly modify the problem setting $\lambda_{\underline{\alpha}} = \Lambda_b > 0,\ \forall \underline{\alpha} \in \mathcal{U}_{i,b} \cup \mathcal{I}_{i,P}$, in Eq. (20) and choosing the constant Λ_b to be smaller than machine accuracy (typically $\Lambda_b = 10^{-20}$). The time index (l) is omitted in this section and superscripts $[n]$ in square brackets are used to identify the solution after n iteration steps.

3.1. The IMSFV procedure

Additional grids and partitions of the integration domain into subdomains (coarse and dual cells) have to be introduced in order to implement the IMSFV procedure. We define the following terminology. Two triplets $\underline{\alpha}$ and $\underline{\beta}$ belonging to the same grid $\mathcal{G}$ are said to be diagonally opposed if $\underline{\alpha} < \underline{\beta}$ and $\underline{\beta} \leqslant \underline{\gamma}$ for all $\underline{\gamma} \in \mathcal{G}$ with $\underline{\alpha} < \underline{\gamma}$. A subset of $\mathcal{G}$, which is itself a grid, is said to be a subgrid of $\mathcal{G}$.

The coarse grid $\bar{\mathcal{P}}$ and the dual grid $\tilde{\mathcal{P}}$ (see Fig. 4) are subgrids of $\mathcal{P}$ and

$$\mathcal{O} = \left\{ \underline{\alpha} :\ \alpha_j \in \mathbb{Z}\ \forall j,\ \underline{x}_{\underline{\alpha}} \in \Omega \right\}, \tag{22}$$

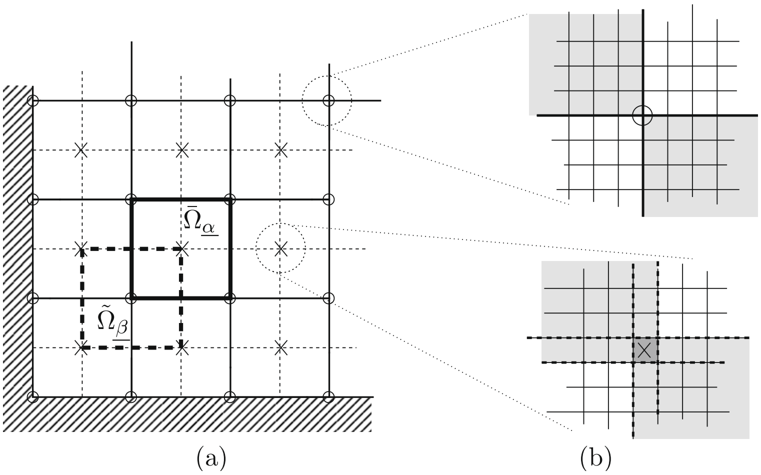
respectively, whereby $\tilde{\mathcal{P}}$ contains its boundary. Let also $\overline{\overline{\mathcal{P}}}$ be the smallest grid containing $\overline{\mathcal{P}} \cup \partial\overline{\mathcal{P}}$. Dual and coarse grids are staggered with respect to each other, i.e. if $\underline{\alpha}_1, \underline{\alpha}_2 \in \overline{\overline{\mathcal{P}}}$ are diagonally opposite and $\underline{\alpha}_1 < \underline{\alpha}_2$, then there exists exactly one $\underline{\beta} \in \tilde{\mathcal{P}}$, so that $\underline{\alpha}_1 \leqslant \underline{\beta} \leqslant \underline{\alpha}_2$. In general, except when $\underline{\beta} \in \partial\tilde{\mathcal{P}}$, strict inequality holds and $\underline{x}_{\underline{\beta}}$ is close to or coincident with the midpoint between $\underline{x}_{\underline{\alpha}_1}$ and $\underline{x}_{\underline{\alpha}_2}$. Analogously, if $\underline{\alpha}_1, \underline{\alpha}_2 \in \tilde{\mathcal{P}}$ are diagonally opposite and $\underline{\alpha}_1 < \underline{\alpha}_2$, there exists exactly one $\underline{\beta} \in \overline{\mathcal{P}}$, so that $\underline{\alpha}_1 < \underline{\beta} < \underline{\alpha}_2$. The original grid $\mathcal{P}$ will also be referred to as fine grid.

Let $\underline{\alpha}_1, \underline{\alpha}_2 \in \tilde{\mathcal{P}}$ be diagonally opposite, with $\underline{\alpha}_1 < \underline{\alpha}_2$, and let $\underline{\beta}$ be the unique element of $\bar{\mathcal{P}}$ satisfying $\underline{\alpha}_1 < \underline{\beta} < \underline{\alpha}_2$, then the coarse cell $\overline{\Omega}_{\underline{\beta}}$ is defined as

$$\overline{\Omega}_{\underline{\beta}} = \bigcup_{\underline{\gamma} \in \mathcal{P}, \underline{\alpha}_1 < \underline{\gamma} < \underline{\alpha}_2} \Omega_{\underline{\gamma}}. \tag{23}$$

Analogously, the dual cell $\widetilde{\Omega}_{\underline{\beta}}$ is given by

$$\widetilde{\Omega}_{\underline{\beta}} = \bigcup_{\underline{\gamma} \in \mathcal{P}, \underline{\alpha}_1 \leqslant \underline{\gamma} \leqslant \underline{\alpha}_2} \Omega_{\underline{\gamma}}, \tag{24}$$



$\bar{\Omega}_{\underline{\alpha}}$
$\tilde{\Omega}_{\underline{\beta}}$
(a)
(b)

Fig. 4. (a) Coarse (solid lines) and dual grid (dashed lines); bold lines highlight the coarse cell $\bar{\Omega}_{\underline{\alpha}}$ (solid) and the dual cell $\tilde{\Omega}_{\underline{\beta}}$ (dashed). Nodes in $\bar{\mathcal{P}}$ are marked by crosses, nodes in $\tilde{\mathcal{P}}$ by circles; (b) enlargement of the corner region of coarse (top) and dual cells (bottom). Thin lines mark the boundaries of fine-grid cells, thick lines the boundaries of coarse and dual cells, respectively. Notice the overlapping of the dual cells in the lower picture. Nodes in $\mathcal{P}$ are the fine-grid-cell centroids.

where $\underline{\alpha}_1, \underline{\alpha}_2 \in \overline{\overline{\mathcal{P}}}$ are diagonally opposite and $\underline{\beta}$ is the only element of $\tilde{\mathcal{P}}$ with $\underline{\alpha}_1 \leqslant \underline{\beta} \leqslant \underline{\alpha}_2$. Notice that different coarse cells share at most faces of their boundary. On the contrary, dual cells share fine-grid cells adjacent to common corners, edges and faces.

Local grids $\bar{\mathcal{P}}_{\underline{\alpha}}$ and $\tilde{\mathcal{P}}_{\underline{\alpha}}$ and their boundary sets $\partial\bar{\mathcal{P}}_{\underline{\alpha}}$ and $\partial\tilde{\mathcal{P}}_{\underline{\alpha}}$ are defined for coarse and dual cells $\overline{\Omega}_{\underline{\alpha}}$ and $\widetilde{\Omega}_{\underline{\alpha}}$ as $\mathcal{P}$ and $\partial\mathcal{P}$ for the integration domain Ω. For example, $\bar{\mathcal{P}}_{\underline{\alpha}}$ is the set of elements of $\mathcal{P}$ contained in the coarse cell $\overline{\Omega}_{\underline{\alpha}}(\bar{\mathcal{P}}_{\underline{\alpha}} = \left\{\underline{\beta} \in \mathcal{P} :\ \Omega_{\underline{\beta}} \subset \overline{\Omega}_{\underline{\alpha}}\right\})$, while $\partial\bar{\mathcal{P}}_{\underline{\alpha}}$ contains the projection of elements of $\bar{\mathcal{P}}_{\underline{\alpha}}$ onto its faces. Finally, for $\underline{\alpha} \in \tilde{\mathcal{P}}$ and $\underline{\beta} \in \tilde{\mathcal{P}}_{\underline{\alpha}}$, we define

$$B_{\underline{\alpha},\underline{\beta}} = \left\{j : \exists v \in \{\pm 1\} \text{ with } \left[(\underline{\beta} + 2v\delta_j) \notin \tilde{\mathcal{P}}_{\underline{\alpha}}\right] \wedge [(\underline{\beta} + 2v\delta_j) \in \mathcal{P}]\right\}. \tag{25}$$

The set $B_{\underline{\alpha},\underline{\beta}}$ is empty for nodes $\underline{\beta} \in \tilde{\mathcal{P}}_{\underline{\alpha}}$, which are not adjacent to the boundary $\partial\widetilde{\Omega}_{\underline{\alpha}}$ of the dual cell $\widetilde{\Omega}_{\underline{\alpha}}$, and contains the indices j of the directions normal to the adjacent dual-cell faces otherwise. As an exception, the direction j is not contained in $B_{\underline{\alpha},\underline{\beta}}$, if the cell face normal to it lies on $\partial\Omega$.

The solution $p^{[n+1]}$ after $n+1$ iteration steps is obtained by superposing the basis functions $\Phi_{\underline{\alpha}}$, $\underline{\alpha} \in \bar{\mathcal{P}}$, and the correction function Ψ, and applying the line-relaxation operator S a prescribed number n_s of times[1]:

$$p' = \sum_{\underline{\alpha} \in \bar{\mathcal{P}}} p'_{\underline{\alpha}} \Phi_{\underline{\alpha}} + \Psi, \tag{26a}$$

$$p^{[n+1]} = S^{n_s}(p'), \tag{26b}$$

Basis functions $\Phi_{\underline{\alpha}}$ are zero outside of the eight dual cells (4 in 2-d) sharing the node $\underline{\alpha} \in \bar{\mathcal{P}}$ as a corner and are computed only once before starting the actual computation. On the contrary, the correction function and the coefficients $p'_{\underline{\alpha}}$ have to be recomputed at each iteration step.

Each basis function $\Phi_{\underline{\alpha}}$, $\underline{\alpha} \in \bar{\mathcal{P}}$, is computed by solving independent problems on each of the dual cells forming its support. Let $\widetilde{\Omega}_{\underline{\beta}}, \underline{\beta} \in \tilde{\mathcal{P}}$, be contained in the support of $\Phi_{\underline{\alpha}}$. The restriction of $\Phi_{\underline{\alpha}}$ to $\widetilde{\Omega}_{\underline{\beta}}$ is then determined by

$$\sum_{j \in B^C_{\underline{\beta},\underline{\gamma}}} \left[\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Phi_{\underline{\alpha}}}{\partial x_j}\right)\right]_{\underline{\gamma}} = 0, \quad \forall \underline{\gamma} \in \tilde{\mathcal{P}}_{\underline{\beta}} \setminus \bar{\mathcal{P}}, \tag{27a}$$

$$(\Phi_{\underline{\alpha}})_{\underline{\gamma}} = 0, \quad \forall \underline{\gamma} \in \tilde{\mathcal{P}}_{\underline{\beta}} \cap \bar{\mathcal{P}},\ \underline{\gamma} \neq \underline{\alpha}, \tag{27b}$$

$$(\Phi_{\underline{\alpha}})_{\underline{\gamma}} = 1, \quad \underline{\gamma} = \underline{\alpha}, \tag{27c}$$

$$\left(\lambda \frac{\partial \Phi_{\underline{\alpha}}}{\partial x_k}\right)_{\underline{\gamma}} = 0, \quad \forall \underline{\gamma} \in \partial\tilde{\mathcal{P}}_{\underline{\beta}} \cap \partial\mathcal{P}, \tag{27d}$$

where $B^C_{\underline{\beta},\underline{\gamma}} = \{1,2,3\} \setminus B_{\underline{\beta},\underline{\gamma}}$ and k in Eq. (27d) is the direction normal to $\partial\Omega$ at the corresponding boundary node. Similarly, the restriction of the correction function Ψ to the any dual cell $\widetilde{\Omega}_\alpha, \underline{\alpha} \in \tilde{\mathcal{P}}$, is determined by

$$\sum_{j \in B^C_{\underline{\alpha},\underline{\gamma}}} \left[\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Psi}{\partial x_j}\right)\right]_{\underline{\gamma}} + \sum_{j \in B_{\underline{\alpha},\underline{\gamma}}} \left[\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial p^{[n]}}{\partial x_j}\right)\right]_{\underline{\gamma}} = R_{\underline{\gamma}}, \quad \forall \underline{\gamma} \in \tilde{\mathcal{P}}_{\underline{\alpha}} \setminus \bar{\mathcal{P}}, \tag{28a}$$

$$(\Psi)_{\underline{\gamma}} = 0, \quad \forall \underline{\gamma} \in \tilde{\mathcal{P}}_{\underline{\alpha}} \cap \bar{\mathcal{P}}, \tag{28b}$$

$$\left(\lambda \frac{\partial \Psi}{\partial x_k}\right)_{\underline{\gamma}} = q_{\underline{\gamma}}, \quad \forall \underline{\gamma} \in \partial\tilde{\mathcal{P}}_{\underline{\alpha}} \cap \partial\mathcal{P}, \tag{28c}$$

where $B^C_{\underline{\alpha},\underline{\gamma}}$ and k are defined as above. Finally, the coarse-grid problem (coarse system) assigning the coefficients $p'_{\underline{\alpha}}$, $\underline{\alpha} \in \bar{\mathcal{P}}$ in Eq. (26a) is derived as a weak formulation of the original problem (19) over each coarse cell by requiring

$$\int_{\overline{\Omega}_{\underline{\alpha}}} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial p'}{\partial x_j}\right) dV = \int_{\partial\overline{\Omega}_{\underline{\alpha}}} \lambda \frac{\partial p'}{\partial x_j} n_j dS = \int_{\overline{\Omega}_{\underline{\alpha}}} R dV, \forall \underline{\alpha} \in \bar{\mathcal{P}}, \tag{29}$$

$\underline{n}$ being the unit outwards-pointing normal vector on $\partial\overline{\Omega}_{\underline{\alpha}}$. Substituting the ansatz (26a) into Eq. (29) one obtains

$$\sum_{\underline{\beta} \in \bar{\mathcal{P}}} p'_{\underline{\beta}} \underbrace{\int_{\overline{\Omega}_{\underline{\alpha}}} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Phi_{\underline{\beta}}}{\partial x_j}\right) dV}_{C_{\underline{\alpha}\underline{\beta}}} = \underbrace{-\int_{\overline{\Omega}_{\underline{\alpha}}} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Psi}{\partial x_j}\right) dV + \int_{\overline{\Omega}_{\underline{\alpha}}} R dV}_{Q_{\underline{\alpha}}}, \quad \forall \underline{\alpha} \in \bar{\mathcal{P}}. \tag{30}$$

Eqs. (27a) and (28a) enforce the Poisson equation (respectively in its homogeneous and inhomogeneous form) exactly in the inner part of each dual cell, but only approximatively at nodes adjacent to dual-cell faces not lying on $\partial\Omega$. At such nodes, derivatives normal to the dual-cell boundary are ignored (Eq. (27a)) or estimated a priori on the basis of the last available

[1] We slightly deviate from the notation used in [1], where $\Phi_{\underline{\alpha}}$ and Ψ are expressed as superposition of their restrictions to dual cells.

iteration step (Eq. (28a)). This way, coupling between different dual cells is suppressed, and localization of the problems for $\Phi_{\underline{\alpha}}$ and Ψ is achieved. On the other hand, the coarse system (30) provides the global coupling needed to avoid performance decay when the dimensions of the problem are increased. Indeed, as highlighted by Hajibeygi et al. [1], the IMSFV procedure can be interpreted as a two-level multi-grid procedure, whereby restriction (from $\mathcal{P}$ to $\bar{\mathcal{P}}$, Eq. (30)) and prolongation (from $\bar{\mathcal{P}}$ to $\mathcal{P}$, Eq. (26a)) are carried out by means of the basis functions $\Phi_{\underline{\alpha}}$. The relevance of the line relaxation operator S^{n_s} in Eq. (26b) for the stability and the efficiency of the IMSFV procedure has also been demonstrated by Hajibeygi et al. [1].

The unique solvability of each localized problem for basis and correction functions can be inferred as follows, if $\lambda \neq 0$ everywhere in Ω. Due to the neglect or approximate estimation of derivatives across the boundary, the solution at node-rows corresponding to the dual-cell edges is uniquely determined by 1-d elliptic problems along the edge with Dirichlet conditions at the corners. Two-dimensional problems with Dirichlet boundary conditions provided by the previously computed solution on the edges are then assigned at dual-cell faces and, in turn, provide Dirichlet conditions for the 3-d problems determining the solution inside each dual cell. Exceptions are given at corners, edges and faces lying on $\partial\Omega$, where the original Neumann conditions are set, but this does not compromise the conclusions. Notice that, since the restriction of basis and correction functions to nodes next to dual-cell edges and faces are always determined by the same 1-d and 2-d problems, no contradiction is found, even if such nodes are shared by different dual cells and each basis or correction function is redundantly determined there by different localized problems (one for each dual cell sharing the specific edge or face).

Let now N_c be the dimension of the system matrix $\mathbf{C} = [C_{\underline{\alpha}\underline{\beta}}]$ for the coarse system (30). If $p'_{\underline{\alpha}} = c \in \mathbb{R},\ \forall \underline{\alpha} \in \bar{\mathcal{P}}$, is the constant distribution on the coarse grid, the corresponding superposition of basis functions according to the first sum on the right-hand side of Eq. (26a) is constant on the fine grid $\mathcal{P}$ and the left-hand side of the coarse system (30) is zero. This shows that constant distributions on $\bar{\mathcal{P}}$ lie in the null space of $\mathbf{C}$ and consequently also that $rank(\mathbf{C}) \leqslant N_c - 1$. If the permeability λ is constant and if coarse cells have moderate aspect ratio, $\mathbf{C}$ is diagonally dominant [17] and therefore $rank(\mathbf{C}) \geqslant N_c - 1$. Continuity considerations allow then to conclude that the same is true also when λ varies moderately. This is the case for the applications we are interested in, so that in the following we may always assume $rank(\mathbf{C}) = N_c - 1$. Numerical evidence provided by Hajibeygi et al. [1] shows that the same may still be true also for cases with large cell aspect ratios or strongly varying permeability.

Since each linear combination of basis functions fulfills the homogeneous Neumann conditions (27d) on the domain boundary $\partial\Omega$, the Gauss theorem for the discrete case can be used to characterize the range of the system matrix $\mathbf{C}$. For any set of values $p'_{\underline{\alpha}}$, $\underline{\alpha} \in \bar{\mathcal{P}}$, on the coarse grid we obtain

$$\sum_{\underline{\alpha}\in\bar{\mathcal{P}}}\sum_{\underline{\gamma}\in\bar{\mathcal{P}}} p'_{\underline{\gamma}} C_{\underline{\alpha}\underline{\gamma}} = \sum_{\underline{\alpha}\in\bar{\mathcal{P}}}\sum_{\underline{\gamma}\in\bar{\mathcal{P}}} p'_{\underline{\gamma}} \int_{\overline{\Omega}_{\underline{\alpha}}} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Phi_{\underline{\gamma}}}{\partial x_j}\right) dV = \sum_{\underline{\gamma}\in\bar{\mathcal{P}}} p'_{\underline{\gamma}} \int_{\Omega} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Phi_{\underline{\gamma}}}{\partial x_j}\right) dV = \sum_{\underline{\gamma}\in\bar{\mathcal{P}}} p'_{\underline{\gamma}} \int_{\partial\Omega} \lambda \frac{\partial \Phi_{\underline{\gamma}}}{\partial x_j} n_j dS = 0, \tag{31}$$

which immediately provides the necessary condition

$$\sum_{\underline{\alpha}\in\bar{\mathcal{P}}} Q_{\underline{\alpha}} = \sum_{\underline{\alpha}\in\bar{\mathcal{P}}}\left[-\int_{\overline{\Omega}_{\underline{\alpha}}} \frac{\partial}{\partial x_j}\left(\lambda \frac{\partial \Psi}{\partial x_j}\right) dV + \int_{\overline{\Omega}_{\underline{\alpha}}} R dV \right] = 0 \tag{32}$$

for the solvability of the system (30). Eq. (32) however is implied by the solvability condition (21), and therefore by Eq. (5) in the Navier–Stokes context, since the correction function Ψ fulfills Eq. (28c). Under the assumption that $rank(\mathbf{C}) = N_c - 1$, we can therefore conclude that the coarse system (30) is always solvable, if Eq. (21) is verified.

3.2. Modifications of the IMSFV procedure

Two reasons are responsible for the bad performance of the IMSFV procedure, when λ approaches zero in portions of Ω. First, the sensitivity of the iterative procedure with respect to the approximations introduced to derive localized problems for basis and correction functions increases. Second, the condition of the coarse system worsens. Both problems are discussed hereafter and modifications to the IMSFV algorithm are introduced to remove them.

The proof of solvability for the localized problems defining basis and correction functions fails in presence of impermeable regions, if portions of dual-cell edges lie in the permeable domain but are disconnected from the respective corners due to the presence of impermeable cells (Fig. 5). As highlighted in Section 3.1, the solution for basis and correction functions along such edges is governed by 1-d elliptic problems, which in the specific case may be restricted to edge-segments lying in the permeable domain, with homogeneous Neumann conditions (no-flow) at the boundary to the impermeable regions. While the right-hand side of the 1-d problems for the basis functions is homogeneous, that for the correction function is not and might be incompatible with the Neumann conditions, in which case no solution can be found. This is the limit case, but when λ approaches zero within nearly impermeable regions, correction and basis functions become increasingly sensitive to the approximations leading to the 1-d elliptic problems. The quality of $\Phi_{\underline{\alpha}}$ and Ψ worsens correspondingly and the convergence behaviour of the iterative procedure is negatively affected. Analogous problems may also appear, if disconnected permeable regions with no connection to the edges are given on dual-cell faces.

Low values of λ may also affect the condition number of the coarse system, when coarse-grid nodes $\underline{\alpha} \in \bar{\mathcal{P}}$ lie within the low-permeability region, but portions of the corresponding coarse cells $\overline{\Omega}_{\underline{\alpha}}$ do not. Fluxes $\lambda\partial\Phi_{\underline{\alpha}}/\partial x_j$ induced by the basis function $\Phi_{\underline{\alpha}}$ decrease proportionally to the value, say λ_{low}, of λ inside the low-permeability region (Fig. 6). Correspondingly, also

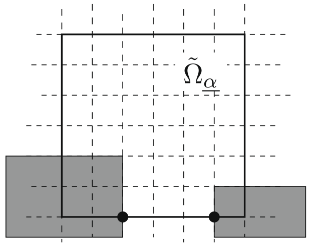


Fig. 5. Disconnected portion of dual-cell edge. Dashed lines are fine-cell boundaries, the solid line is the boundary of the dual cell. Impermeable regions are shaded.

entries $C_{\underline{\gamma},\underline{\alpha}},\ \underline{\gamma} \in \mathcal{P}$, of the coarse-system matrix **C** tend to zero ($C_{\underline{\gamma},\underline{\alpha}} \sim \lambda \partial \Phi_{\underline{\alpha}} / \partial x_j \sim \mathcal{O}(\lambda_{low})$). At the same time, contributions of neighbouring cells $\underline{\beta} \in \bar{\mathcal{P}},\ \underline{\beta} \neq \underline{\alpha}$, to the flux balance for $\overline{\Omega}_{\underline{\alpha}}$ do not have to decrease in the same measure. Coefficients differing by several orders of magnitude appear in the matrix **C**, thus enhancing the sensitivity of the system to numerical cancellation. In the limit, when λ becomes zero, the rank of **C** decreases by the number of coarse-grid nodes lying within the impermeable region.

The following modifications have been introduced into the IMSFV procedure in order to remove the difficulties discussed above:

(1) The correction function Ψ is either fully discarded or its restriction to each dual cell $\widetilde{\Omega}_{\underline{\alpha}}$ is computed by replacing Eq. (28a) at edge nodes $\underline{\gamma} \in \mathcal{B}_{\underline{\alpha}} = \left\{ \underline{\gamma} \in \tilde{\mathcal{P}}_{\underline{\alpha}} : \mathcal{B}_{\underline{\alpha},\underline{\gamma}} \neq \emptyset \right\}$ with more robust Dirichlet conditions.
(2) A clipped permeability field $\hat{\lambda} = max\{\lambda, \hat{\Lambda}\},\ \hat{\Lambda} > 0$ is employed in place of λ to compute the basis functions (i.e. in (27d) but not in (30)).
(3) The iterative procedure is reimplemented as a residuum-correction scheme using the IMSFV step as a preconditioner.

Correspondingly, the following algorithmic steps are carried out in the modified IMSFV procedure to proceed from $p^{[n]}$ to $p^{[n+1]}$:

i. Compute the residuum $r^{[n]}$ in the fulfillment of Eq. (19) for the approximate solution $p^{[n]}$.
ii. Compute the correction function $\delta\Psi$ by solving Eq. ()(28) on each dual cell $\widetilde{\Omega}_{\underline{\alpha}}$ with $\delta\Psi$ and $r^{[n]}$ in place of Ψ and R, respectively. Consider thereby homogeneous Neumann conditions on $\partial\Omega$ and replace Eq. (28a) at nodes $\underline{\gamma} \in \mathcal{B}_{\underline{\alpha}}$ with the homogeneous Dirichlet condition

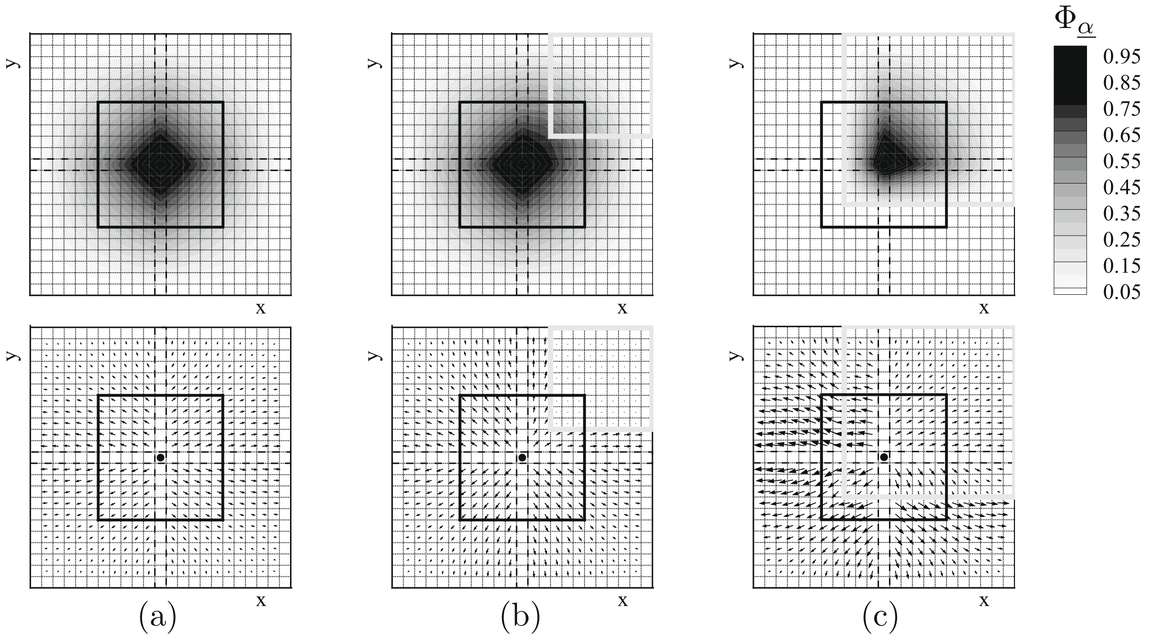


Fig. 6. Isocontours of 2-d basis function $\Phi_{\underline{\alpha}}$ (top) and fluxes $\lambda \partial \Phi_{\underline{\alpha}} / \partial x_j$ (bottom): no impermeable region (a), node $\underline{\alpha}$ lies outside impermeable region (b), node $\underline{\alpha}$ lies inside impermeable region (c). Dashed and solid black lines are the boundaries of dual and coarse cells, respectively. The node $\underline{\alpha}$ is marked by a black dot and impermeable regions by thick gray lines. The permeability within the impermeable region is $\lambda = 10^{-3}$. Fluxes in (c) are upscaled by factor 10^3.

$$(\delta\Psi)_{\underline{\gamma}} = 0, \quad \forall\underline{\gamma} \in \mathcal{B}_{\underline{\alpha}}. \tag{33}$$

iii. Compute the coarse-grid correction $\delta p'_{\underline{\alpha}}, \underline{\alpha} \in \bar{\mathcal{P}}$, solving the coarse system (30) with $\delta p'_{\underline{\alpha}},\ \delta\Psi$ and $r^{[n]}$ in place of $p'_{\underline{\alpha}}, \Psi$ and R, respectively. Also replace $\Phi_{\underline{\alpha}}$ with basis functions $\hat{\Phi}_{\underline{\alpha}}$ computed considering the clipped permeability field $\hat{\lambda}$.

iv. Reconstruct the fine-scale solution and apply line relaxation:

$$p' = p^{[n]} + \sum_{\underline{\alpha}\in\bar{\mathcal{P}}} \delta p^{[n]}_{\underline{\alpha}} \hat{\Phi}_{\underline{\alpha}} + \delta\Psi, \tag{34a}$$

$$p^{[n+1]} = S^{n_s}(p'). \tag{34b}$$

Two main elements contribute to ensure convergence and efficiency of the IMSFV procedure. They are already present in the original version but become even more evident in the modified one. First, the coarse system (30) enforces the global coupling throughout the integration domain. Second, local refinement of the solution, achieved by line-relaxation and through the computation of the correction function Ψ, enforces convergence at the fine-grid level. Indeed, the computation of Ψ with the boundary condition (33) can be interpreted as a domain-decomposition-based relaxation step and becomes conceptually equivalent to the line-relaxation step.

The correction-residuum formulation used in the modified procedure allows for more flexibility in the definition of correction and basis functions, which according to Eq. (34a) are only used to represent the incremental correction term and not the full solution as in (26a). Thanks to that, basis functions computed for a modified permeability field and the correction function considering the robust boundary condition (33) can be used without renouncing fine-grid convergence. Indeed, the correction function may even be discarded, in which case local adaptation to the fine-grid features of permeability and source terms is enforced solely by line relaxation (in contrast, line relaxation can not be easily removed due to its importance to ensure the stability of the procedure [1]).

Analogies between the IMSFV procedure and a two-stage multi-grid method [1] are even more evident for the modified version presented above. Basis functions represent thereby the key element to define both the restriction and the prolongation operators. The fact that basis functions provide an excellent tool to formulate the coarse-grid problem taking into account the fine-grid features of the permeability field is indeed the basic idea of multi-scale approaches [2]. In addition, we point out their suitability for reconstructing the fine-grid solution according to Eq. (34a). Also there, basis functions account for fine-grid features of the permeability, thus minimizing the interference between global step and local refinement (e.g. they do not induce fluxes through impermeable regions, cf. Fig. 6(b)). Furthermore, the prolongation operator correctly maps the null space of the coarse system problem onto the null space of the fine-grid problem (in both cases the subspaces of the constant functions, see Section 3.1). This is a necessary prerequisite, since null-space elements are left undetermined by the coarse-grid problem and would corrupt the solution, if not mapped into the null-space of the fine-grid problem. All positive features of the basis functions are preserved, when they are defined considering a clipped permeability field $\hat{\lambda}$ with a proper choice of the lower bound $\hat{\Lambda}$.

Under the assumption that λ can become arbitrarily small but not zero, the following conclusions can be drawn about the solvability and the condition number of the modified IMSFV procedure. Fine-scale problems for basis and correction functions are always diagonally dominant and therefore well conditioned, if the inhomogeneous terms R and q in Eq. (19) may be assumed to be everywhere at most of the same order of magnitude as the local permeability (no large sources within impermeable regions). Equations for nodes in the permeable and impermeable regions then differ by a scaling factor proportional to the local permeability, but are equally uncritical from a numerical point of view, if a floating point representation is used for real numbers. Difficulties arising in connection with 1-d or 2-d problems on disconnected dual-cell edges or faces have been removed by setting condition (33) for the correction function and by considering the clipped permeability field $\hat{\lambda}$ for the basis functions. Finally, the solvability of the coarse-grid problem can be inferred from the discussion of Section 3.1, considering that the lower bound $\hat{\Lambda}$ for the clipped permeability field $\hat{\lambda}$ defining the basis functions can always be chosen (but in praxis this is usually not necessary) in such a way that variations of $\hat{\lambda}$ within the integration domain are arbitrarily small. Numerical evidence on this subject is provided in Section 4.1.

3.3. Combined iteration for the IMSFV procedure and IIM

We conclude the presentation of the Navier–Stokes solver with IIM capabilities by describing the procedure used to enforce Eq. (18) at immersed boundaries by means of iterative recomputation of the extrapolated velocities $u^{(l+1)}_{i,\underline{\alpha}}$ at the ghost-nodes $\underline{\alpha} \in \mathcal{I}_{i,P}$. The iterative update of $u^{(l+1)}_{i,\underline{\alpha}}$ is thereby integrated into the IMSFV loop, thus avoiding the implementation of additional loops and preserving good convergence rates also for the combined problem. The following algorithmic steps are considered to advance the velocity field from $\underline{u}^{(l)}$ to $\underline{u}^{(l+1)}$:

i. Integrate the momentum Eq. (11) to evaluate $u^{(l+1)}_{i,\underline{\beta}}$ at nodes $\underline{\beta}$ involved in the extrapolation stencils defining $u^{(l+1)}_{i,\underline{\alpha}}$ at $\underline{\alpha} \in \mathcal{I}_{i,P}$. Consider the pressure from the previous time step to evaluate the pressure gradient in Eq. (11).

ii. Compute $u^{(l+1)}_{i,\underline{\alpha}}, \underline{\alpha} \in \mathcal{I}_{i,P}$, by extrapolation.

iii. Compute the right-hand side R of Eq. (19a), which depends on $u^{(l+1)}_{i,\underline{\alpha}},\ \underline{\alpha} \in \mathcal{I}_{i,P}$, through the boundary condition (18).

iv. Compute p' according to the modified IMSFV procedure (Eq. (34a)).
v. Apply a single line-relaxation step in each spatial direction.
vi. Update $u^{(l+1)}_{i,\underline{\alpha}}$, $\underline{\alpha} \in \mathcal{I}_{i,P}$, and R as in steps i–iii considering the last available pressure field.
vii. Return to step v until the number n_s of relaxation steps is reached.
viii. Return to step iv until the convergence criterion is fulfilled.
ix. Evaluate $u^{(l+1)}_{i,\underline{\alpha}}$ in $\underline{\alpha} \in \mathcal{U}_{i,f} \setminus \mathcal{I}_{i,P}$ from Eq. (11).

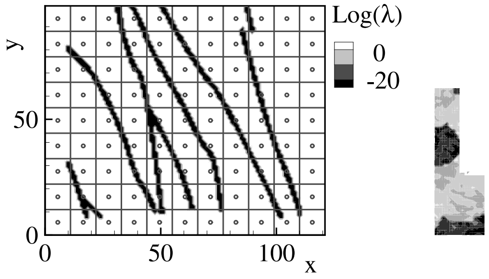
Extrapolated values at ghost nodes $\underline{\alpha} \in \mathcal{I}_{i,P}$ are updated after each application of the line-relaxation operator S (steps v to vii). This way, boundary conditions at immersed boundaries are first accurately forced onto the solution locally, and then their effect is spread over the whole integration domain by the following global step (step iv). The IMSFV procedure is well adapted to this strategy for two reasons. First, local improvement can be cheaply achieved within the local relaxation step. Second, the global step does not interfere with that since the basis functions $\Phi_{\underline{\alpha}}$, which are used for prolongation, do not induce fluxes through the boundaries of impermeable regions (cf. Fig. 6(b)). Numerical evidence for these statements is provided in Section 4.3. Notice, that extrapolated values at nodes $\underline{\alpha} \in \mathcal{I}_{i,U}$ need not be modified within the iterative loop, since they are only needed to evaluate the right-hand side of the momentum equation, which is not recomputed within the loop.

4. Numerical tests

4.1. *Modified IMSFV for elliptic problems with impermeable regions*

To investigate the performance of the original IMSFV procedure and of its modified version, the solution of the elliptic Eq. (19) is considered for two demanding distributions of the permeability λ. In the test case named Impermeable-Layer, λ is equal to 1 everywhere, except within thin impermeable layers where it is set to $\lambda = 10^{-20}$. This is exactly the problem one would be confronted with in the IIM context, if solid bodies with the shape of the impermeable layers were put into the flow domain. On the other hand, the test case Spe-10 is derived from a classical problem for reservoir simulations (Spe-10 bottom layer [18]) and is similar to a test case considered by Hajibeygi et al. [1]. In contrast to the first test case, the permeability assumes more than two values in the range between 2×10^{-3} and 2×10^{4}. Homogeneous Neumann boundary conditions are set in both cases on the whole boundary $\partial\Omega$ and concentrated sources and sinks are located at opposite corners of the integration domain. The fine grid for case Impermeable-Layer includes 121 × 99 fine cells, which are uniformly divided into 11 × 9 coarse cells. Case Spe-10 includes 220 × 60 fine and 22 × 6 coarse cells. The permeability distributions and the coarse grids are depicted in Fig. 7. In both test cases some coarse-grid nodes lay within low-permeability or impermeable regions.

The bad convergence behaviour of the original IMSFV procedure for problems with impermeable regions is illustrated in Fig. 8. The maximum difference between iterative (p) and exact solution (p_f) is plotted as a function of the iteration number for computations considering different numbers n_s of relaxation steps and clipped permeability fields $\check{\lambda}$. Replacing the original permeability field λ with $\check{\lambda}$ modifies the problem to be solved and, therefore, also the reference solution p_f. According



to the original IMSFV procedure, the basis functions are evaluated considering the permeability $\check{\lambda}$ for the problem at hand. Convergence rates deteriorate in both test cases, when the lower bound $\check{\Lambda}$ is decreased, and the procedure diverges if $\check{\lambda}$ varies over more than two (for $n_s = 5$) or three (for $n_s = 20$) orders of magnitude within the integration domain. We mention that the original IMSFV procedure converges for the unclipped Spe-10 case, if $n_s \gtrsim 250$ (see [1]), while computations considering double-precision for the representation of real numbers break down independently of n_s, if the original permeability distribution of case Impermeable-Layer is taken into account (due to the near singularity of the 1-d problems for basis and correction functions on dual-cell edges).

Fig. 9 shows results from computations with the modified IMSFV procedure. Thereby, different values were considered for the lower bound $\hat{\Lambda}$ of the permeability field $\hat{\lambda}$ used in the computation of the basis functions $\hat{\Phi}_{\underline{\alpha}}$. In contrast to the computations of Fig. 8 the target problem in not modified and considers the original permeability distribution. In both test cases convergence is fastest for $\hat{\Lambda} \approx 10^{-4}\lambda_{max}$, where λ_{max} is the largest value of λ in Ω. Further reduction of $\hat{\Lambda}$ does not improve the convergence rates but has a rather negative effect on the error of the converged solution. For excessively small values of $\hat{\Lambda}$, the iterative procedure diverges. Differences in the behaviour of computations with and without correction function are moderate.

The instability resulting from the bad quality of the correction function at disconnected edge or face portions of the dual cells has been suppressed in Fig. 9 either by introducing the Dirichlet condition (33) or by discarding the computation of Ψ. Differences in the behaviour for different $\hat{\Lambda}$ are therefore entirely due to the quality of the basis functions and of the resulting coarse system (30). On one hand, decreasing $\hat{\Lambda}$ leads to basis functions, which better account for the actual permeability distribution, thus improving the quality of the coarse-scale solution and enhancing the convergence rate. On the other hand, lowering $\hat{\Lambda}$ negatively affects the condition of the coarse system (30) for the reasons discussed in Section 3.2. Amplified rounding errors are introduced at each iteration by the global step and can be compensated only partially by line relaxation. We also point out, that taking $\hat{\Lambda} = 10^{-1}$ in case Impermeable-Layer gives an iterative procedure, in which the global step, i.e. an important part of correction computation, nearly ignores the presence of the low permeability layers. This is similar to what is done in the IIM context, when the pressure correction is computed ignoring the presence of immersed bodies. The resulting performance is clearly not satisfactory.

The condition number $K = \|\mathbf{C}\|_2 / \|\mathbf{C}^{-1}\|_2$ of the coarse system (30) is plotted in Fig. 10 as a function of the lower bounds $\check{\Lambda}$ and $\hat{\Lambda}$ for the permeability $\check{\lambda}$ defining the problem to be solved and for the permeability $\hat{\lambda}$ considered for the computation of the basis functions, respectively. We computed K after removing the singularity of the matrix connected to the constant solution and after normalizing the lines of $\mathbf{C}$, so that the largest element of each line has absolute value 1. This second step is admissible since we may assume that the right-hand side of each equation is at most of the same order of magnitude as the largest coefficient at the left-hand side (see also the analogous discussion on the solvability of the problems for correction and basis functions at the end of Section 3.2). Lines marked by square symbols show the condition number K as a function of the lower bound $\check{\Lambda}$, when no lower bound $\hat{\Lambda}$ is set to define the basis functions. Lines with upward pointing triangles show the dependence of K on $\hat{\Lambda}$ for computations considering the original permeability distribution λ. Finally, lines marked by downwards pointing triangles show the variation of K as a function of $\check{\Lambda}$, while $\hat{\Lambda}$ is kept equal to $10^{-4}\lambda_{max}$. Dashed lines in Fig. 10(b) are computed considering the stretched permeability field $\lambda_{str} = \lambda^{3/2}$, where λ is the permeability of case Spe-10. Our goal was to define a problem, which is as demanding as the case Immpermeable-Layer with respect to the ratio $\lambda_{max}/\lambda_{min}$, but preserves the complex topology of case Spe-10.

Conclusions from Fig. 10 are quite straightforward. The condition number K for the original IMSFV procedure scales as $1/\hat{\Lambda}$. If $\hat{\Lambda}$ is set to a reasonable level, low values within the actual permeability field only have a moderate influence on K,

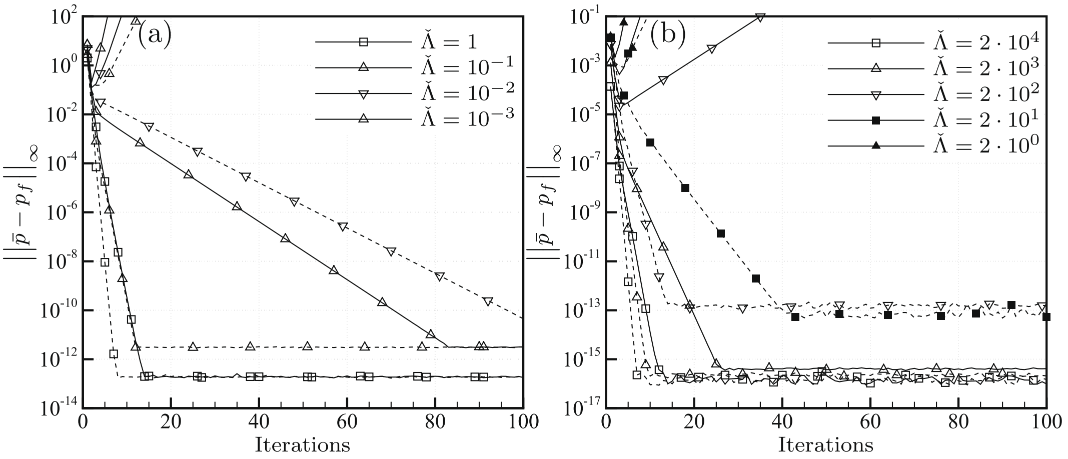
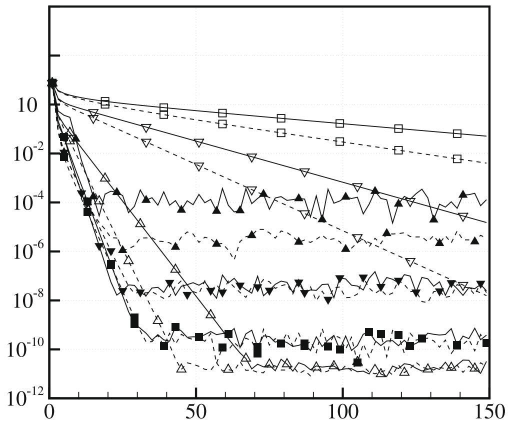


Fig. 8. Convergence history of the original IMSFV procedure for the cases Impermeable-Layer (a) and Spe-10 (b) with clipped permeability distributions $\check{\lambda} = \max\{\lambda, \check{\Lambda}\}$ and decreasing values of $\check{\Lambda}$. Solid and dashed lines refer to computations with $n_s = 5$ and $n_s = 20$, respectively.



so that solvability and accuracy are preserved also in the presence of impermeable regions. We emphasize that, when the computations of Fig. 9 were repeated with enhanced accuracy of the floating point representation (i.e. using the type long double in place of double to represent real numbers in our c++ code) the error at convergence decreased for both test cases by nearly the same factor as the machine accuracy improved. This provides conclusive evidence that the errors shown in Fig. 9 are indeed due to the bad condition of the coarse system, and not to conceptual limitations of the iterative procedure.

4.2. Navier–Stokes solver and IIM

Two test cases are considered to validate the algorithm for the solution of the Navier–Stokes equations and the IIM implementation of the immersed boundaries. In test case Volume–Force, a term $\underline{F}(t,\underline{x})$ is added to the right-hand side of the momentum Eq. (3a), such that the velocity field $\underline{u}_{ex}$ derived from the stream function

$$\phi_{ex}(t,\underline{x}) = \frac{1}{4}[\cos(\pi x_1) - 1][\cos(\pi x_2) - 1]\cos(\omega t), \quad \omega \in \mathbb{R} \tag{35a}$$

and the pressure field

$$p_{ex}(t,\underline{x}) = cos(\pi x_1)cos(\pi x_2)\cos(\omega t) \tag{35b}$$

exactly fulfill the modified equations in $\Omega = [-1,1] \times [-1,1]$ for $Re = 40$. The circle with center at the origin and radius $r = 0.2$ is then treated as a solid body and non-homogeneous Dirichlet boundary conditions are derived from the analytical

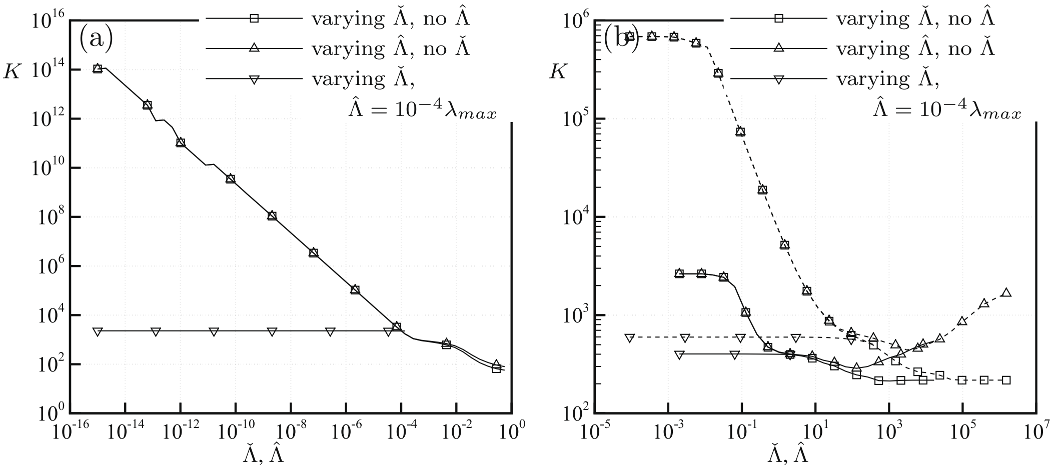


Fig. 10. Condition number $K = \|\mathbf{C}\|_2/\|\mathbf{C}^{-1}\|_2$ of the coarse system (30) as a function of the lower bounds $\check{\Lambda}$ and $\hat{\Lambda}$ for the cases Impermeable-Layer (a) and Spe-10 (b). Dashed lines in (b) refer to computations for the stretched permeability field $\lambda_{str}(\underline{x}) = \lambda(\underline{x})^{3/2}$.

solution (35a). In test case Cylinder the classical flow around a circular cylinder is considered for $Re = 20$, $Re = 40$ and $Re = 100$. For the first two values of the Reynolds number the wake behind the cylinder remains symmetric and steady, while in the third case an unsteady Karman vortex street develops. The integration domain is $\Omega = [-10, 20] \times [-10, 10]$ and the constant velocity $\underline{u}_\infty = (1\ 0)^T$ is imposed at the inflow boundary ($x_1 = -10$). The diameter of the cylinder is $d = 1$ and its center is located at the origin. In all cases, the permeability within the solid bodies (cf. Eq. (20)) is $\lambda_b = 10^{-20}$ and the lower-bound for the computation of basis functions is $\hat{\Lambda} = 10^{-3}$.

The accuracy of the spatial discretization is verified in Fig. 11(a) on the basis of case Volume–Force. Maximum errors in the numerical solution for velocity and pressure are plotted as functions of the grid spacing $\Delta x_1 = \Delta x_2$ for steady computations ($\omega = 0$). Second order accuracy is achieved for all variables. Furthermore reference data from a computation without the immersed body demonstrate that immersed boundaries do not induce a significant enhancement of the error. Fourth-order accuracy with respect to the time step Δt can be observed in Fig. 11(b), where the maximum error at time $t = 10$ is shown for an unsteady computation with $\omega = 80\pi$ and $\Delta x_1 = \Delta x_2 = 0.05$. Thereby, a large circular frequency ω was considered in order to increase the time-integration error with respect to the spatial discretization error. For small values of Δt, the spatial-discretization error becomes dominant and the curves level up. Errors for computations without solid bodies are almost identical and therefore not reported.

Results for the case Cylinder at $Re = 20$ and $Re = 40$ are shown in Fig. 12. The grid spacing is $\Delta x_1 \approx \Delta x_2 \approx 0.0725$ for both computations. As expected, the recirculation region behind the cylinder grows for increasing Reynolds number, but remains symmetric and steady. A quantitative comparison of the present results with data from the literature is provided in Table 1

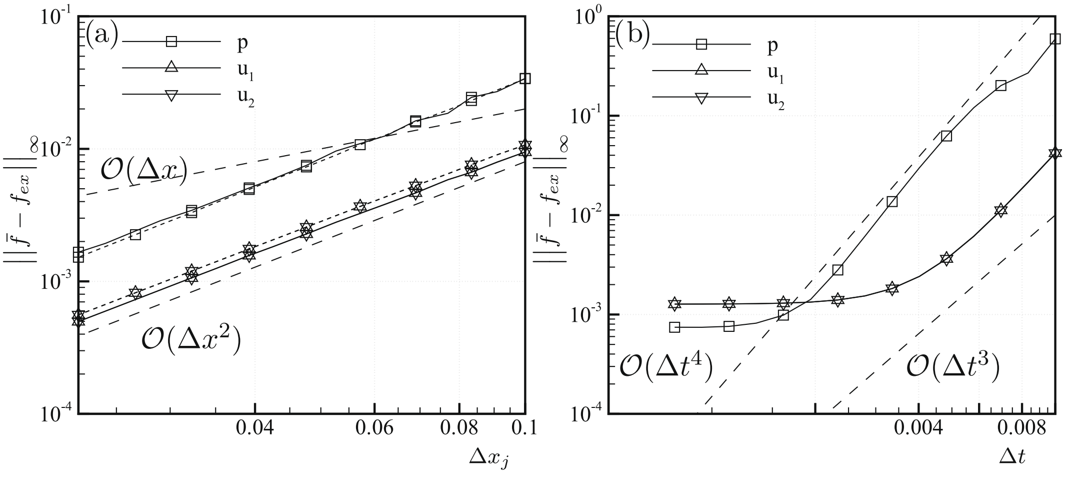


Fig. 11. Maximum error in the case Volume–Force for steady computations as a function of the grid spacing $\Delta x_1 = \Delta x_2$ (a) and for unsteady computations ($\omega = 80\pi$) with $\Delta x_1 = \Delta x_2 = 0.05$ as a function of the time step Δt (b). Dashed lines in figure (a) refer to computations without immersed body.

and in Fig. 13 for the drag coefficient C_d, for the main geometrical parameters of the wake and for the distribution of pressure and vorticity on the cylinder surface. Our results lie within the spread of the reference data. In particular, almost perfect agreement can be observed with respect to computational results by Linnick and Fasel [10], who considered an integration domain of nearly the same size as ours. On the other hand, Xu [19] considered a slightly smaller integration domain, which explains the deviations from the present results for C_D and for the pressure distribution in Fig. 13.

The grid spacing for the computation of case Cylinder at $Re = 100$ was reduced to $\Delta x_1 \approx \Delta x_2 \approx 0.0483$ and the wake instability was triggered at the begin of the simulation by means of a pulse-like volume force. The resulting drag and lift coefficients are plotted in Fig. 14(a) as functions of time. Mean values, amplitude and frequency of their oscillations are compared

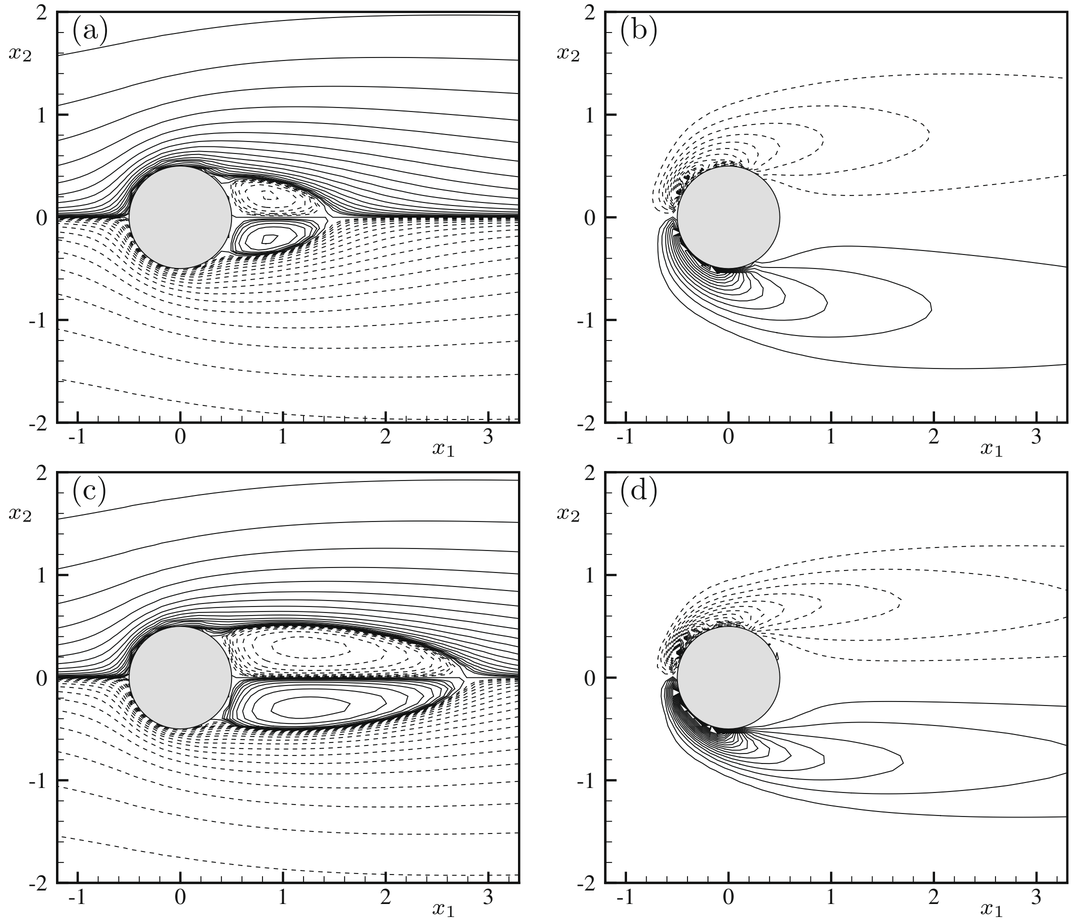


Fig. 12. Isocontours of the stream-function (a, c, logarithmic spacing) and spanwise vorticity ω_3 (b, d, $\Delta\omega_3 = 0.5$) for case Cylinder at $Re = 20$ (top) and $Re = 40$ (bottom). Dashed lines correspond to negative values.

Table 1
Geometrical parameters of the steady wake and drag coefficient for case Cylinder at $Re = 20$—40 : $(x_T, 0)$ and $(x_S, \pm y_S)$ are the stagnation points at the trailing edge and at the center of the recirculation bubbles, respectively. Results by Coutanceau and Bouard [20] are extrapolated values for the limit case of an infinite domain. Results by Linnick and Fasel [10] refer to their computation with smaller integration domain.

	$Re = 20$				$Re = 40$			
	x_T	x_S	$2 \cdot y_S$	C_D	x_T	x_S	$2 \cdot y_S$	C_D
Tritton [21]	–	–		2.16	–		–	1.59
Coutanceau and Bouard [20]	1.43	0.83	0.47	–	2.63	1.26	0.59	–
Braza et al. [22]	1.45	–	–	2.17	2.56	–	–	1.57
Linnick and Fasel [10]	1.43	0.86	0.43	2.16	2.73	1.21	0.59	1.61
Xu [19]	1.43	0.86	0.43	2.23	2.74	1.22	0.60	1.66
Present	1.41	0.85	0.42	2.17	2.76	1.20	0.62	1.62

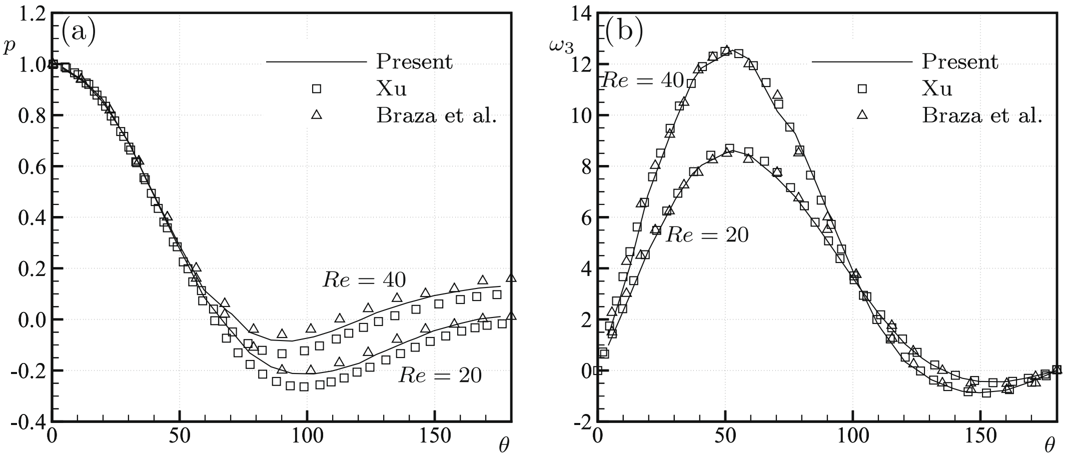
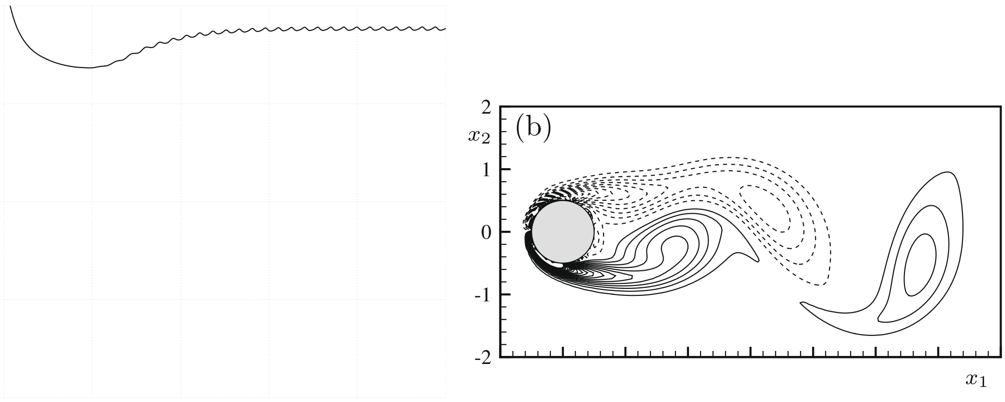


Fig. 13. Pressure (a) and vorticity distribution on the cylinder surface at $Re = 20$ and $Re = 40$ (b). Reference data are taken from Xu [19] and Braza et al. [22].



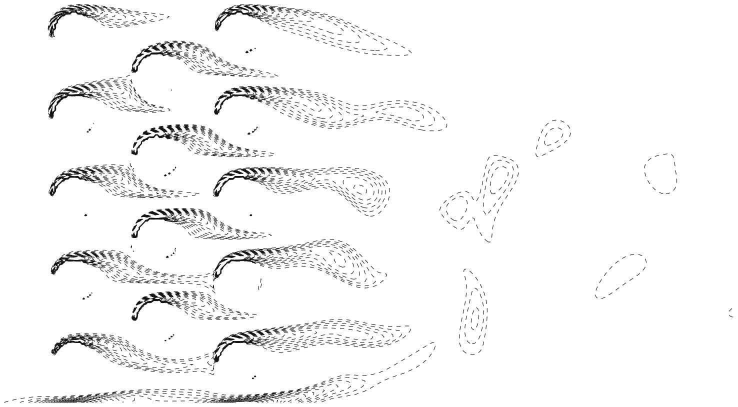
in Table 2 with data from the literature, finding again satisfactory agreement. The well developed Karman street is visualized by means of vorticity isocontours in Fig. 14(b).

In conclusion a flow visualization for a problem with complex geometry is shown in Fig. 15 in order to highlight the potential of IIM. We considered the flow in a 2-d channel of height $H = 5$, crossed by a cluster of circular cylinders with diameter $d = 1$. The grid spacing was $\Delta x_1 \approx \Delta x_2 \approx 0.0483$ as for case Cylinder at $Re = 100$. The wake of the upstream cylinder is clearly reduced and stabilized by the presence of the downstream ones. Vortical structures similar to those observed in the Karman vortex street develop only downstream of the last row of cylinders.

4.3. Combined iteration for Poisson solver and immersed boundaries

We conclude by verifying the convergence rate of the IMSFV procedure for the Poisson equation combined with the iterative updating of the boundary conditions at immersed boundaries. Computations were run for the unsteady test case Volume–Force, considering different spatial discretization levels and a large frequency for the forcing term ($\omega = 80\pi$) in order to enhance the work-load for the Poisson solver. The time step was $\Delta t = 0.0025$ for all computations.

Results from three different sets of computations are reported to evaluate different updating strategies. Solid lines in Figs. 16(a) and (c) were obtained by following the algorithm described in Section 3.3. Dashed lines in the same figures refer to computations, in which the immersed boundary conditions were updated only once at the end of each IMSFV iteration step (steps vi and vii from Section 3.3 are swapped). Finally, results in Figs. 16(b) and (d) were obtained without updating the boundary condition (step vi is discarded). This last case represents the optimum from the view point of the Poisson solver, since all constraints are kept constant during the IMSFV iteration, but the boundary condition (18) is enforced only approx-



imatively. Convergence Γ rates are defined as the average variation of the basis-10 logarithm of the residuum r over an IMSFV iteration.

The provided results confirm what has been said in Section 3.3. Updating the boundary condition (18) after each line-relaxation step (solid lines in figures a and c) enforces local adaptation of the solution, which can be efficiently accounted for in the subsequent global step. Convergence rates are therefore not significantly reduced with respect to the case without updating. This is particularly true when the number of smoothing steps n_s is increased, since local adaptation is then improved. On the contrary, if immersed boundary conditions are updated only once in each IMSFV iteration (dashed lines in figures a and c), local adaptation is achieved only insufficiently and therefore the global step can not work efficiently. Convergence rates decrease by up to one order of magnitude with respect to the optimum case. The fact that the immersed-boundary treatment represents the bottle neck limiting the overall convergence is confirmed by the observation that convergence rates do not respond to parameters like the number of line-relaxation steps or the coarse-cell size, which are known to have an influence on the IMSFV procedure [1].

The grid spacing, i.e. the fine-cell number N_j in Fig. 16 does not effect convergence rates. Indeed, thanks to the global coupling provided by the coarse system, IMSFV is not sensitive to the problem size [1] and modifying the grid spacing, while keeping constant the cell aspect ratio, only leads to a rescaling of the coefficient in Eq. (16) but does not modify their structure. Oscillations in the curves of Fig. 16 are rather due to different relative positions of grid nodes and immersed boundaries, which lead to more or less favourable interpolation stencils. We refer to Hajibeygi et al. [1] for an investigation of the role played by the cell aspect ratio.

Since immersed boundary conditions do not massively influences the performance of the IMSFV procedure in the IIM context, we may refer to Section 4.1 to evaluate the dependence of converge rates on the complexity of the immersed-body geometry. High convergence rates as those observed in Fig. 16 ($\Gamma \approx 0,5{-}1.5$) are only achieved for simple geometries, while the permeability distribution considered in Section 4.1 are probably worse than the most complex geometry one might reasonably consider for a simulation with IIM. We obtain $\Gamma \approx 0,2{-}0.25$ for case Impermeable Layer and $\Gamma \approx 0,06{-}0.2$ for case Spe-10. Notice that the Reynolds number has no influence on the convergence rate since it appears only in the right hand side of Eq. (16).

The IIM computations of Section 4.2 have been carried out with $n_s = 5$ relaxation steps within each IMSFV iteration and considering 5×5 fine cells in each coarse cell. In agreement with what has been just said, the resulting convergence rates were $\gamma \approx 1$ for case Cylinder, i.e. for a geometry similar to the one considered in this section, and $\Gamma \approx 0.2$ for the computation of the channel flow with multiple immersed cylinders. Benchmark values from the literature could be found only in Mark and van Wachem [12], who report $\Gamma \approx 0.15$ for the computation of the flow around a sphere. To little is said, however, about the details of the iterative procedure to allow a comparison of the cost-performance ratio.

5. Conclusions

The IMSFV method by Hajibeygi et al. [1] for the iterative solution of elliptic problems has been generalized to consider problems with large impermeable regions and integrated into the IIM procedure by Peller et al. [11] as an efficient solver for the pressure Poisson equation resulting from the incompressible Navier–Stokes equations. Since high accuracy is required in the Navier–Stokes context, only the iterative feature of the IMSFV procedure is considered, ignoring its capability of providing good approximations of the solution already after one single iteration.

The main differences of the modified IMSFV formulation with respect to the original one [1] are the introduction of approximated basis functions obtained by considering a clipped permeability field, and the replacement of the approximated boundary conditions for the correction function at dual-cell edges with more robust Dirichlet conditions. Alternatively, the correction function may also be discarded. Finally, a residuum-correction iteration scheme, in which the IMSFV step acts as a preconditioner, has to be considered, since the fine-scale solution can not be represented as a superposition of the modified correction and basis functions. The combination of all these modifications successfully removes the stability and accuracy limitations afflicting the original procedure in cases in which the permeability varies over more than 3 orders of magnitude throughout the integration domain.

Analogies pointed out by Hajibeygi et al. [1] between the IMSFV procedure and a two-level multi-grid approach with large upscaling factors between fine and coarse grids are further enhanced by the modifications introduced in this paper. Two main elements appear as fundamental in the modified procedure. First, the multi-grid step enforces fast coupling of the solution over the whole integration domain, thus preventing performance decay when enlarging the dimensions of the problem. Second, local relaxation, mainly achieved by line-relaxation but also through the computation of the correction function, refines the solution locally, thus ensuring convergence at the fine-grid level. We underline that the key element of the multigrid step is represented by the basis functions derived from the multi-scale context. They allow the formulation of the coarse-grid problem (restriction) and the interpolation of its solution onto the fine grid (prolongation), accounting in both cases for the fine-grid features of the permeability field. Furthermore, when used for prolongation, basis functions correctly map the null space of the coarse-grid problem into that of the fine-grid problem.

The role of the correction function has been reduced in the modified IMSFV procedure, becoming analogous to that of line-relaxation. For the considered test cases, no deterioration of the performance was observed when the computation of the correction function was discarded, but we do not exclude that for specific permeability distributions the kind of local relaxation they induce might be complementary to that produced by line relaxation. Furthermore, what has just been said is only valid for the IMSFV algorithm considered as an iterative procedure and has no relevance to evaluate the effect of correction function on the quality of the solution after one single iteration, i.e. on the final solution for non-iterative multi-scale approaches.

The Poisson equation governing the pressure in the IIM formulation by Peller et al. [11] has been recast into a form amenable to efficient solution with the IMSFV procedure. Since a direct enforcement of the full constraints at immersed boundaries would seriously disrupt the structure of the problem, an iterative procedure has been introduced, according to which the pressure field and the velocity at proper ghost nodes are updated in an alternating manner. The IMSFV procedure is used as a preconditioner to compute the pressure correction at each iteration. By assigning zero permeability to regions occupied by solid bodies, it allows to account approximatively for boundary conditions already at that stage. The gain with respect to standard preconditioner ignoring the presence of immersed bodies is significant.

The fact that the iterative handling of ghost nodes within the IIM context can be efficiently integrated within the iterative loop of the IMSFV procedure underlines the adequateness of the latter for such applications. The reason for this positive outcome lies in the structure of the IMSFV procedure, and in particular in the distinction between global steps and local relaxation. Thanks to that the solution can be adapted to the boundary constraint, locally and at low computational cost, before spreading the influence of the local improvements into the whole domain by means of the global step. Local, cheap, iterative adaptation replaces the direct enforcement of the non-standard discretization at nodes next to the boundary.

We do not exclude that other elliptic solvers, may also be efficiently combined with the immersed boundaries approach. Adaptation however is probably not trivial as it was not for the IMSFV method. For example, simple multi-grid methods fail in presence of impermeable regions and also the procedure by Alcouffe et al. [23], explicitly developed for that purpose, might diverge if the topology of impermeable regions is too complex. Excellent robustness could be shown for the IMSFV method in this respect and its compatibility with the treatment of immersed boundary conditions is a further positive aspect. Finally, we point out that the extension of the IMSFV method to cases with impermeable regions is of relevance also for other applications, where elliptic equations with similar structure have to be solved, e.g. getting back to the original field of application of IMSFV, for the simulation of flows in porous media.

References

[1] H. Hajibeygi, G. Bonfigli, M.A. Hesse, P. Jenny, Iterative multiscale finite-volume method, J. Comput. Phys. 277 (2008) 8604–8621.

[2] T.Y. Hou, X.H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput. Phys. 134 (1997) 169–189.

[3] P. Jenny, S.H. Lee, H.A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, J. Comput. Phys. 187 (2003) 47–67.

[4] I. Lunati, P. Jenny, Multi-scale finite-volume method for compressible multiphase flows in porous media, J. Comput. Phys. 216 (2006) 616–636.

[5] L.J. Durlofsky, Y. Efendiev, V. Ginting, An adaptive local-global multiscale finite volume element method for two-phase flow simulations, Adv. Water Res. 30 (2007) 576–588.

[6] R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.

[7] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.

[8] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.

[9] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.

[10] M. Linnick, H. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.

[11] N. Peller, A. Le Duc, F. Tremblay, M. Manhart, High-order stable interpolation for immersed boundary methods, Int. J. Numer. Meth. Fluids 52 (2006) 1175–1193.

[12] A. Mark, B. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, J. Comput. Phys. 227 (2008) 6660–6680.

[13] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[14] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[15] R.L. Sani, J. Shen, O. Pironneau, P.M. Gresho, Pressure boundary condition for the time-dependent incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 50 (2006) 673–682.

[16] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.

[17] M.A. Hesse, B. Mallison, S. Lee, H.A. Tchelepi, Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations, SIAM Multiscale Model. Simul. 583 (2008) 229–272.

[18] M.A. Christie, M.J. Blunt, 10th SPE comparative solution project: a comparison of upscaling techniques, in: SPE 66599, February 2001.

[19] S. Xu, The immersed interface method for simulating prescribed motion of rigid objects in an incompressible flow, J. Comput. Phys. 227 (2008) 5045–5071.

[20] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. steady flow, J. Fluid Mech. 79 (1977) 231–256.

[21] D.J. Tritton, Experiments on the flow past a circular cylinder at low reynolds numbers, J. Fluid Mech. 6 (1959) 547–567.

[22] M. Braza, P. Chassaing, H. Ha Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79–130.

[23] R.E. Alcouffe, Achi Brandt, J.E. Dendy Jr., J.W. Painter, The multi-grid method for the diffusion equation with strongly discontinuous coefficients, SIAM J. Sci. Stat. Comput. 2 (4) (1981) 430–454.